

BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG
PHÂN HIỆU TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA CÔNG NGHỆ THÔNG TIN

---o0o---



KHAI PHÁ DỮ LIỆU

Xây dựng mô hình phân loại hoa sử dụng bộ dữ liệu 102

Category Flower Dataset

Sinh viên thực hiện	Lê Quốc Vinh
	Đào Nguyễn Duy Bảo
	Huỳnh Minh Nghĩa
Lớp	S25-64CNTT
Giảng viên hướng dẫn	ThS. Vũ Thị Hạnh

Tp.Hồ Chí Minh, tháng 10, năm 2025

Mục lục

Chương 1. Tổng quan	3
1.1. Lý do chọn đề tài	3
1.2. Mục tiêu của đề tài	3
1.3. Giới thiệu bài toán và bộ dữ liệu	5
Chương 2: Cơ sở lý thuyết và công cụ sử dụng	6
2.1. Kiến thức nền tảng về khai phá dữ liệu (Data Mining)	6
2.2. Kiến thức về Học sâu (Deep Learning) trong phân loại ảnh	9
Chương 3: Quy trình và xử lý dữ liệu	11
3.1. Mô tả bộ dữ liệu	11
3.2. Tiền xử lý dữ liệu	14
3.2.1 Chuẩn hóa kích thước và định dạng ảnh	14
3.2.2 Chuẩn hóa giá trị pixel	14
3.2.3 Tăng cường dữ liệu (Data Augmentation)	15
3.2.4 Chia tập dữ liệu	16
3.3. Tăng cường dữ liệu và trực quan hóa kết quả	18
Chương 4: Xây dựng và huấn luyện mô hình	21
4.1. Cấu trúc mô hình EfficientNetB0	21
4.1.1 Xây dựng cơ bản cho mô hình	21
4.1.2 Tham số huấn luyện	23
4.1.3 Kết quả huấn luyện	26
4.1.4 Đánh giá mô hình	30
4.2 Cấu trúc mô hình MobileNetV2	32
4.2.1 Xây dựng cơ bản cho mô hình	32
4.2.2 Tham số huấn luyện	33
4.2.3 Kết quả huấn luyện	36
4.2.4 Đánh giá mô hình	39
4.3 Cấu trúc mô hình CNN	40
4.3.1 Xây dựng cơ bản cho mô hình	40
4.3.2 Tham số huấn luyện	43
4.3.3 Kết quả huấn luyện	45
4.3.4 Đánh giá mô hình	48
Chương 5 : So sánh giữa các mô hình	51

5.1 Tổng hợp kết quả huấn luyện	51
5.2 So sánh chi tiết quá trình huấn luyện	51
5.3. So sánh ma trận nhầm lẫn giữa các mô hình.....	53
5.4. Kết luận	54
Chương 6: Hướng phát triển.....	55
6.1. Nâng cao hiệu năng mô hình.....	55
6.2 Mở rộng phạm vi ứng dụng.....	56
6.3 Cải thiện chất lượng dữ liệu	56
Chương 7: Tổng kết.....	58

LỜI MỞ ĐẦU

Trong kỷ nguyên số hóa hiện nay, dữ liệu bùng nổ dưới nhiều hình thức khác nhau, trong đó dữ liệu hình ảnh chiếm một tỷ trọng đáng kể và chứa đựng nguồn thông tin vô cùng phong phú. Việc trích xuất tri thức hữu ích từ nguồn dữ liệu khổng lồ này đã trở thành một trong những mục tiêu hàng đầu của khoa học máy tính. Môn học Khai phá dữ liệu (Data Mining) cung cấp các phương pháp và kỹ thuật thiết yếu để phát hiện các mẫu, xu hướng và xây dựng các mô hình dự đoán từ dữ liệu thô, đóng vai trò then chốt trong việc giải quyết các bài toán thực tiễn phức tạp.

Một trong những bài toán kinh điển và đầy thách thức trong lĩnh vực thị giác máy tính và khai phá dữ liệu là phân loại hình ảnh (Image Classification). Nhiệm vụ này đòi hỏi máy tính phải "hiểu" và gán nhãn chính xác cho một bức ảnh dựa trên nội dung trực quan của nó.

Bài toán này trở nên đặc biệt phức tạp khi phải phân loại giữa các lớp (categories) có độ tương đồng cao. Bộ dữ liệu "102 Category Flower Dataset" do nhóm Visual Geometry Group (VGG) thuộc Đại học Oxford xây dựng là một ví dụ điển hình. Bộ dữ liệu này bao gồm 102 loài hoa khác nhau, phổ biến tại Vương quốc Anh, với tổng số hơn 8.000 hình ảnh. Đặc điểm của bộ dữ liệu này là sự đa dạng lớn về hình thái ngay trong cùng một loài (biến thiên về tư thế, ánh sáng, tỷ lệ) và sự tương đồng cao giữa các loài khác nhau, đặt ra một thách thức đáng kể cho bất kỳ mô hình phân loại nào.

Với mục tiêu áp dụng các kiến thức đã học từ môn Khai phá dữ liệu vào một bài toán thực tế, nhóm chúng em đã chọn đề tài: "Xây dựng mô hình phân loại hoa sử dụng bộ dữ liệu 102 Category Flower Dataset".

Trong khuôn khổ bài báo cáo này, chúng em sẽ trình bày toàn bộ quy trình thực hiện, bao gồm các bước: thu thập và tiền xử lý dữ liệu, khám phá và trích xuất đặc trưng (feature extraction), lựa chọn và huấn luyện mô hình (bao gồm các kỹ thuật học máy

truyền thống và các phương pháp học sâu - Deep Learning), cuối cùng là đánh giá hiệu suất của mô hình. Qua đó, báo cáo không chỉ nhằm mục đích đạt được kết quả phân loại tốt nhất có thể mà còn mong muốn cung cấp một cái nhìn sâu sắc về cách thức các kỹ thuật khai phá dữ liệu được vận dụng để giải quyết một bài toán nhận diện hình ảnh phức tạp.

Chương 1. Tổng quan

1.1. Lý do chọn đề tài

Môn học Khai phá dữ liệu (Data Mining) cung cấp các phương pháp luận và công cụ để phát hiện các mẫu (patterns) ẩn, các mối liên hệ và xây dựng các mô hình dự đoán từ dữ liệu thô. Một trong những bài toán cốt lõi và phổ biến nhất trong khai phá dữ liệu là bài toán phân lớp (Classification). Khi áp dụng vào dữ liệu hình ảnh, bài toán này được gọi là Phân loại hình ảnh (Image Classification) — một tác vụ nền tảng cho vô số ứng dụng thực tế như nhận diện khuôn mặt, xe tự hành, chẩn đoán bệnh qua ảnh y khoa, hay tự động gắn thẻ ảnh.

Trong bối cảnh đó, việc nghiên cứu và áp dụng các kỹ thuật khai phá dữ liệu tiên tiến, đặc biệt là các phương pháp Học sâu (Deep Learning), để giải quyết một bài toán phân loại hình ảnh cụ thể là vô cùng cần thiết. Đề tài này chọn đối tượng là phân loại các loài hoa, một bài toán thuộc dạng "phân loại chi tiết" (fine-grained classification). Đây là một thử thách thực tế, đòi hỏi mô hình không chỉ phân biệt được các đối tượng khác biệt rõ rệt (như hoa và xe cộ) mà còn phải nhận diện được những khác biệt tinh vi giữa các lớp rất giống nhau (như các loài hoa khác nhau).

1.2. Mục tiêu của đề tài

Từ những vấn đề đã đặt ra, đề tài Xây dựng mô hình phân loại hoa sử dụng bộ dữ liệu 102 Category Flower Dataset được thực hiện nhằm hướng đến các mục tiêu chính sau đây:

1. Nghiên cứu và tìm hiểu:

Tìm hiểu về bài toán phân loại hình ảnh chi tiết (fine-grained image classification) và các thách thức đặc thù của nó (sự tương đồng cao giữa các lớp, sự biến thiên lớn trong cùng một lớp).

Phân tích, khám phá (EDA) và hiểu rõ đặc điểm của bộ dữ liệu "102 Category Flower Dataset".

2. Vận dụng kiến thức:

Áp dụng quy trình chuẩn của một dự án khai phá dữ liệu, bao gồm các bước: thu thập, tiền xử lý dữ liệu, trích xuất đặc trưng, xây dựng mô hình và đánh giá.

Vận dụng các kỹ thuật Học sâu (Deep Learning) tiên tiến, đặc biệt là Mạng nơ-ron tích chập (CNN) và kỹ thuật Học chuyển giao (Transfer Learning), để giải quyết bài toán.

3. Xây dựng mô hình:

Xây dựng thành công một mô hình học máy có khả năng nhận diện và phân loại chính xác 102 loài hoa từ hình ảnh đầu vào.

Thử nghiệm và so sánh hiệu quả của một số kiến trúc mô hình (ví dụ: VGG, ResNet, EfficientNet...) để tìm ra phương án tối ưu cho bộ dữ liệu này.

4. Đánh giá và phân tích:

Sử dụng các độ đo (metrics) chuẩn như Accuracy, Precision, Recall, F1-Score và Ma trận nhầm lẫn (Confusion Matrix) để đánh giá hiệu suất của mô hình một cách khách quan trên tập kiểm thử (test set).

Phân tích các trường hợp mô hình dự đoán sai (error analysis) để hiểu rõ hạn chế của mô hình và đề xuất các hướng cải thiện.

1.3. Giới thiệu bài toán và bộ dữ liệu

Bài toán được đặt ra trong báo cáo này là xây dựng một mô hình khai phá dữ liệu có khả năng nhận đầu vào là một bức ảnh về một loài hoa và dự đoán chính xác tên của loài hoa đó từ 102 loài hoa khác nhau.

Để thực hiện nhiệm vụ này, nhóm chúng em sử dụng bộ dữ liệu "102 Category Flower Dataset" do Nhóm Hình học Trục quan (Visual Geometry Group - VGG) tại Đại học Oxford phát triển. Đây là một bộ dữ liệu chuẩn (benchmark) nổi tiếng, được thiết kế chuyên biệt cho bài toán phân loại hoa chi tiết. Bộ dữ liệu bao gồm 8.189 hình ảnh được chia thành 102 loài hoa phổ biến ở Vương quốc Anh, với số lượng ảnh mỗi lớp dao động từ 40 đến 258.

Thách thức chính của bộ dữ liệu này nằm ở hai đặc điểm:

1. Độ tương đồng cao giữa các lớp (High inter-class similarity): Nhiều loài hoa khác nhau nhưng lại có hình dáng, màu sắc rất giống nhau, gây khó khăn cho việc phân biệt.
2. Độ biến thiên lớn trong cùng một lớp (High intra-class variation): Hình ảnh của cùng một loài hoa có thể rất khác nhau do sự thay đổi về ánh sáng, góc chụp, kích thước, hậu cảnh phức tạp và các giai đoạn phát triển (nụ, nở rộ).

Chính vì những thách thức này, "102 Category Flower Dataset" là một đối tượng lý tưởng để áp dụng và đánh giá hiệu quả của các kỹ thuật khai phá dữ liệu và học máy hiện đại.

Chương 2: Cơ sở lý thuyết và công cụ sử dụng

2.1. Kiến thức nền tảng về khai phá dữ liệu (Data Mining)

Khai phá dữ liệu (Data Mining) là một bước quan trọng trong quy trình phân tích dữ liệu tổng thể, tập trung vào việc khám phá tri thức ẩn chứa trong các tập dữ liệu lớn. Thông qua việc áp dụng các kỹ thuật thống kê, học máy, trí tuệ nhân tạo và trực quan hóa dữ liệu, quá trình này giúp con người phát hiện ra các mẫu, mối liên hệ, xu hướng hoặc hành vi tiềm ẩn mà bằng mắt thường hay phương pháp thủ công khó nhận thấy được.

Khác với việc chỉ “lưu trữ” hay “xử lý” dữ liệu, khai phá dữ liệu mang mục tiêu chuyển dữ liệu thô thành thông tin hữu ích, từ đó hỗ trợ quá trình ra quyết định, dự đoán và tối ưu hóa. Ví dụ, trong thương mại điện tử, khai phá dữ liệu giúp gợi ý sản phẩm cho khách hàng (Recommendation System); trong y tế, nó hỗ trợ phát hiện sớm bệnh dựa trên dữ liệu bệnh án; trong giáo dục, giúp dự đoán sinh viên có nguy cơ bỏ học hoặc đạt thành tích cao.

Trong thời đại dữ liệu bùng nổ, khi khối lượng dữ liệu sinh ra mỗi ngày từ mạng xã hội, cảm biến IoT, giao dịch điện tử, v.v... tăng theo cấp số nhân, Data Mining trở thành cầu nối quan trọng trong chuỗi giá trị dữ liệu (Data Value Chain):

Thu thập → Lưu trữ → Xử lý → Khai phá → Phân tích → Ra quyết định.

Ngày nay, cùng với sự phát triển mạnh mẽ của Trí tuệ nhân tạo (AI), đặc biệt là Học máy (Machine Learning) và Học sâu (Deep Learning), khai phá dữ liệu không còn dừng lại ở các thuật toán thống kê đơn giản mà đã phát triển thành một lĩnh vực liên ngành, nơi máy tính có khả năng tự học, tự thích nghi và cải thiện độ chính xác qua từng lần huấn luyện mô hình.

Mặc dù có mối liên hệ chặt chẽ, Khai phá dữ liệu (Data Mining) và Học máy (Machine Learning) là hai khái niệm khác nhau về phạm vi và mục tiêu. Có thể hiểu ngắn gọn:

- Khai phá dữ liệu là một quy trình tổng thể nhằm phát hiện tri thức từ dữ liệu.
- Học máy là một nhánh công cụ bên trong Data Mining, chuyên về xây dựng mô hình tự động học từ dữ liệu.

Tiêu chí	Khai phá dữ liệu (Data Mining)	Học máy (Machine Learning)
Mục tiêu chính	Tìm kiếm, khám phá và mô tả tri thức ẩn trong dữ liệu.	Xây dựng mô hình toán học có khả năng học từ dữ liệu và dự đoán hoặc ra quyết định tự động.
Phạm vi hoạt động	Bao gồm toàn bộ quy trình CRISP-DM: hiểu nghiệp vụ → thu thập → xử lý → mô hình hóa → đánh giá → triển khai.	Tập trung chủ yếu vào giai đoạn mô hình hóa, huấn luyện thuật toán để học mối quan hệ giữa đầu vào và đầu ra.
Kỹ thuật & công cụ thường dùng	Truy vấn dữ liệu (SQL), thống kê mô tả, trực quan hóa, luật kết hợp (Apriori), phân cụm (K-means), phân loại (Decision Tree), hồi quy,...	Các thuật toán học có giám sát (Supervised: Regression, SVM, Random Forest), không giám sát (Unsupervised: Clustering, PCA), bán giám sát, Reinforcement Learning,...

Mức độ tự động hóa	Cần sự can thiệp và phán đoán của con người, đặc biệt trong khâu xử lý và diễn giải kết quả.	Mô hình có khả năng tự động học từ dữ liệu và cải thiện hiệu suất qua quá trình huấn luyện.
Đầu ra cuối cùng	Tri thức, mẫu, mối quan hệ hoặc luật có thể giải thích và áp dụng trong nghiệp vụ.	Mô hình dự đoán, phân loại hoặc ra quyết định, thường là “hộp đen” (black-box model).
Ứng dụng điển hình	Khai phá giỏ hàng (Market Basket Analysis), phân tích khách hàng (Customer Segmentation), phát hiện gian lận, khai thác văn bản.	Dự đoán giá cổ phiếu, nhận dạng khuôn mặt, xử lý ngôn ngữ tự nhiên (NLP), xe tự hành,...
Vai trò trong mối quan hệ tổng thể	Là toàn bộ quy trình “khai thác” tri thức từ dữ liệu.	Là một công cụ giúp “máy tính học” trong giai đoạn mô hình hóa của Data Mining.
Tiêu chí	Khai phá dữ liệu (Data Mining)	Học máy (Machine Learning)
Mục tiêu chính	Tìm kiếm, khám phá và mô tả tri thức ẩn trong dữ liệu.	Xây dựng mô hình toán học có khả năng học từ dữ liệu và dự đoán hoặc ra quyết định tự động.

2.2. Kiến thức về Học sâu (Deep Learning) trong phân loại ảnh

Học sâu (Deep Learning) là một nhánh của học máy (Machine Learning) sử dụng các mạng nơ-ron nhân tạo (Artificial Neural Networks) có nhiều tầng ẩn (hidden layers) để mô phỏng hoạt động xử lý thông tin của não người. Mục tiêu của học sâu là tự động học ra các đặc trưng phức tạp từ dữ liệu đầu vào, thay vì cần con người thiết kế thủ công như trong các phương pháp truyền thống.

Trong bài toán phân loại ảnh (Image Classification), học sâu – đặc biệt là mạng nơ-ron tích chập (Convolutional Neural Network – CNN) – đóng vai trò then chốt. CNN có khả năng học ra các đặc trưng hình học, màu sắc, và cấu trúc không gian trong ảnh thông qua cơ chế tích chập (convolution) và kết hợp không gian (pooling). Nhờ vậy, mô hình có thể phát hiện các đặc trưng như đường biên, hoa văn, hình khối, và cuối cùng là nhận diện đối tượng trong ảnh.

Một mạng CNN cơ bản thường bao gồm:

- **Lớp tích chập (Convolutional Layer):** Dùng các bộ lọc (kernel) để trích xuất đặc trưng cục bộ.
- **Lớp kích hoạt (Activation Layer):** Áp dụng hàm phi tuyến (ReLU, Sigmoid, Tanh) giúp mô hình học được các quan hệ phức tạp.
- **Lớp gộp (Pooling Layer):** Giảm kích thước không gian và tránh overfitting bằng cách chọn đặc trưng nổi bật nhất.
- **Lớp kết nối đầy đủ (Fully Connected Layer):** Kết hợp các đặc trưng đã học để đưa ra kết quả phân loại cuối cùng.

Trong các bài toán phức tạp, CNN được mở rộng thành các kiến trúc sâu hơn như VGG, ResNet, Inception, EfficientNet, hay Vision Transformer (ViT). Các kiến trúc này được huấn luyện trên tập dữ liệu lớn như ImageNet, giúp mô hình có khả năng nhận dạng đặc trưng tổng quát.

Đối với đề tài *phân loại hoa sử dụng bộ dữ liệu 102 Category Flower Dataset*, nhóm áp dụng Transfer Learning (học chuyển giao) – một kỹ thuật quan trọng trong học sâu. Thay vì huấn luyện mô hình từ đầu, ta tận dụng trọng số đã học từ mô hình được huấn luyện trước (pretrained model) trên tập dữ liệu lớn (như ImageNet) và tinh chỉnh (fine-tune) lại trên bộ dữ liệu hoa. Cách tiếp cận này giúp:

- Giảm thời gian huấn luyện và tài nguyên tính toán.
- Tăng độ chính xác khi dữ liệu huấn luyện hạn chế.
- Giữ được khả năng khái quát hóa của mô hình.

Trong đề tài phân loại 102 loài hoa – vốn có mức độ tương đồng cao giữa các lớp – việc áp dụng Deep Learning mang lại ưu thế vượt trội so với các phương pháp học máy truyền thống (như SVM hay Random Forest). Các tầng CNN và kỹ thuật Transfer Learning giúp mô hình học được đặc trưng tinh vi của hình dáng, cánh hoa, màu sắc, và cấu trúc của từng loài, từ đó tăng khả năng nhận diện chính xác ngay cả trong điều kiện ánh sáng, góc chụp và phong nền khác nhau.

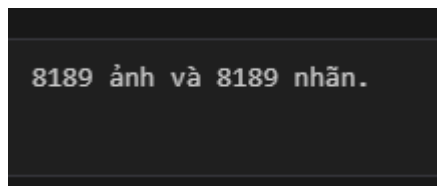
Chương 3: Quy trình và xử lý dữ liệu

3.1. Mô tả bộ dữ liệu

Trong đề tài này, nhóm sử dụng bộ dữ liệu “102 Category Flower Dataset” do Nhóm Hình học Trục quan (Visual Geometry Group – VGG) thuộc Đại học Oxford phát triển. Đây là một trong những bộ dữ liệu tiêu chuẩn nổi tiếng trong lĩnh vực *phân loại hình ảnh chi tiết (fine-grained image classification)*, thường được dùng để đánh giá khả năng nhận dạng đối tượng có hình thái tương tự nhau.

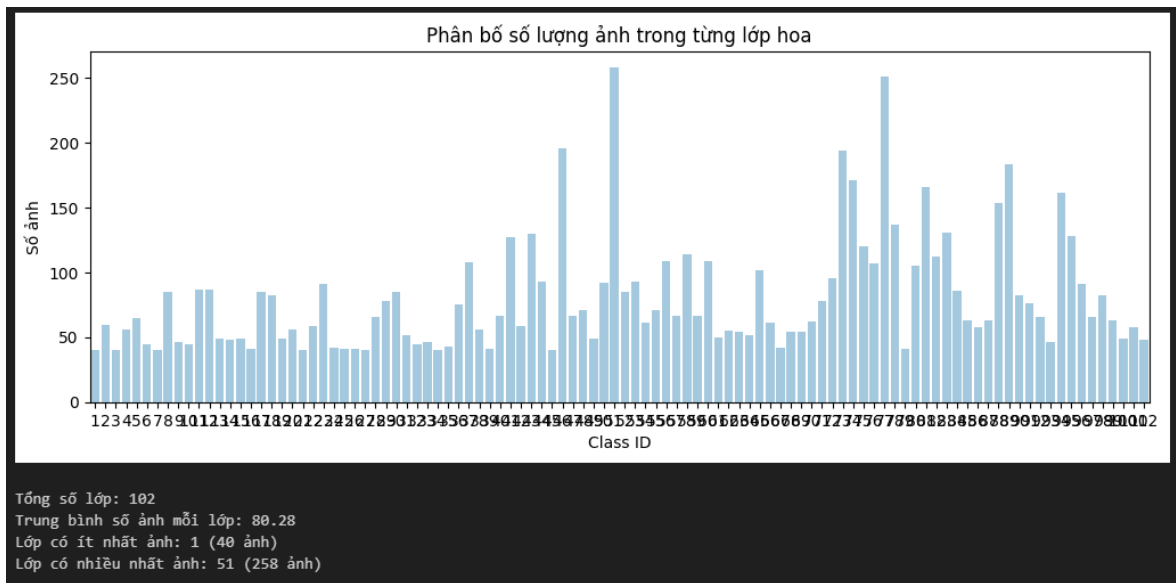
Link: [Phân loại hoa - Flower Classification](#)

Bộ dữ liệu bao gồm 8,189 hình ảnh thuộc 102 loài hoa khác nhau, phổ biến tại Vương quốc Anh. Mỗi loài hoa có số lượng ảnh không đồng đều, dao động từ 40 đến 258 ảnh. Tất cả ảnh đều ở định dạng JPEG, độ phân giải cao (trung bình khoảng 500×500 pixel) và được thu thập trong nhiều điều kiện ánh sáng, góc chụp, và phong nền khác nhau.



Đặc điểm và cấu trúc dữ liệu

- **Số lượng lớp:** 102 (loài hoa).
- **Số lượng ảnh tổng:** 8 189.
- **Định dạng tệp:** JPEG (.jpg).



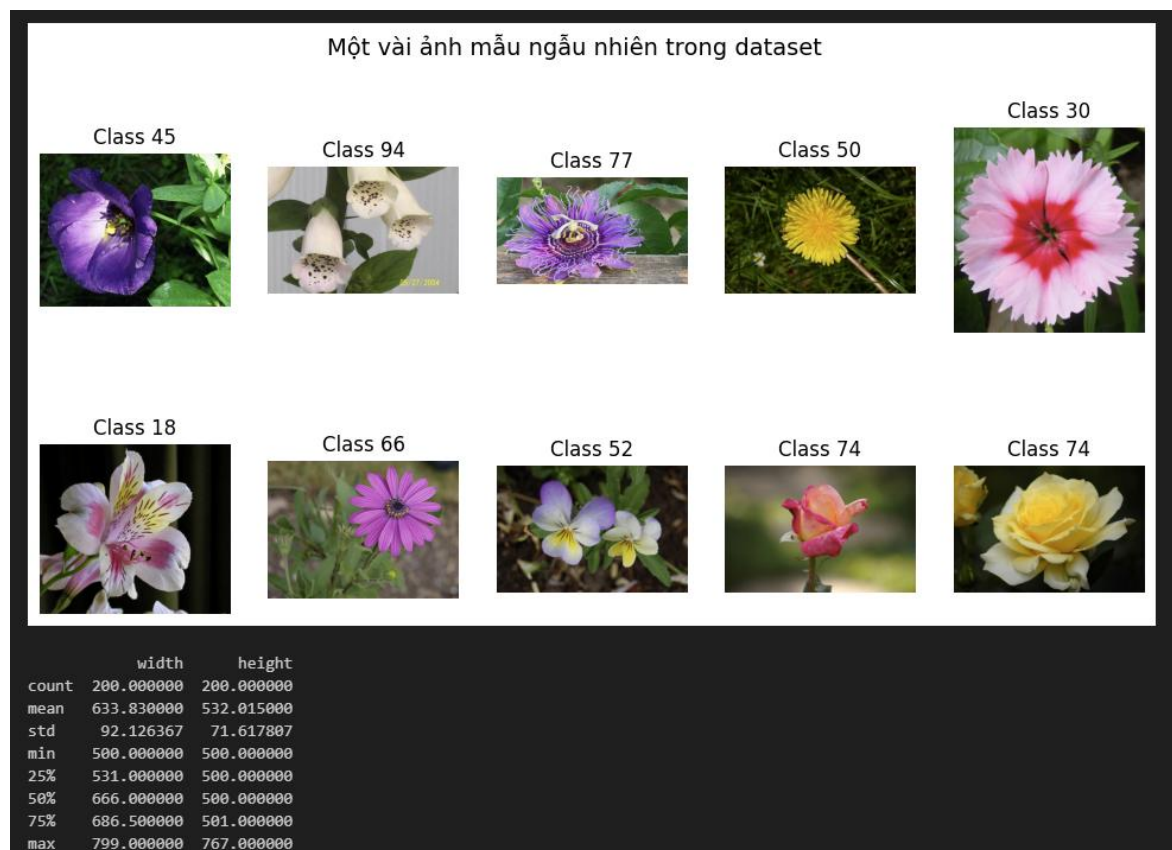
Các ảnh trong bộ dữ liệu được nhóm theo ID và điều này giúp dễ dàng ánh xạ giữa ảnh và nhãn khi huấn luyện mô hình.

Đặc thù và thách thức của bộ dữ liệu

- **Độ tương đồng cao giữa các lớp (High Inter-class Similarity):**
Nhiều loài hoa khác nhau nhưng có hình dáng và màu sắc rất giống nhau (ví dụ: Tulip và Lily), gây khó khăn cho mô hình trong việc phân biệt.
- **Độ biến thiên lớn trong cùng một lớp (High Intra-class Variation):**
Ảnh cùng một loài hoa có thể rất khác nhau do thay đổi ánh sáng, góc chụp, phông nền hoặc giai đoạn nở.
- **Kích thước ảnh và tỷ lệ không đồng nhất:**
Ảnh được chụp từ nhiều nguồn khác nhau, có độ phân giải và tỷ lệ khung hình không đều, cần được chuẩn hóa trước khi đưa vào mạng nơ-ron.
- **Phông nền phức tạp:**
Một số ảnh hoa có nền rừng, đất hoặc vật thể khác, làm tăng độ nhiễu và gây

sai lệch cho mô hình nếu không tiền xử lý cẩn thận.

Sau đây là 1 số ảnh ngẫu nhiên được lấy trong dataset:



Cùng với nó là các thông số để xem width và height của các file ảnh được in ra ở phía dưới.

Những yếu tố này khiến “102 Category Flower Dataset” trở thành bộ dữ liệu chuẩn để đánh giá khả năng phân biệt tinh tế (fine-grained recognition) của các mô hình học sâu.

Mục tiêu sử dụng bộ dữ liệu trong đề tài:

- Huấn luyện một mô hình phân loại ảnh có thể nhận dạng chính xác 102 loài hoa

- VGG16, MobileNetV2, CNN để xây dựng mô hình để phân tích và khai phá dữ liệu hợp lý
- Phân tích đặc trưng và đưa ra nhận xét về những loài hoa mà mô hình thường nhầm lẫn với nhau.

3.2. Tiền xử lý dữ liệu

Để mô hình học sâu có thể huấn luyện hiệu quả, dữ liệu hình ảnh đầu vào cần phải được xử lý đồng nhất về kích thước, định dạng và giá trị pixel. Quá trình tiền xử lý dữ liệu (data preprocessing) đóng vai trò cực kỳ quan trọng, giúp làm sạch và chuẩn hóa dữ liệu trước khi đưa vào giai đoạn huấn luyện. Trong đề tài này, nhóm thực hiện tiền xử lý dựa trên quy trình chuẩn của khai phá dữ liệu và theo các thao tác trong notebook.

3.2.1 Chuẩn hóa kích thước và định dạng ảnh

Do ảnh trong bộ dữ liệu có kích thước và tỉ lệ không đồng nhất (một số ảnh vuông, số khác chữ nhật, độ phân giải chênh lệch lớn), nhóm tiến hành resize toàn bộ ảnh về cùng kích thước 224×224 pixel.

Kích thước này được chọn vì đây là đầu vào mặc định của nhiều mô hình học sâu được huấn luyện sẵn trên ImageNet, đảm bảo khả năng tương thích khi áp dụng kỹ thuật học chuyển giao.

Quá trình chuyển đổi được thực hiện tự động thông qua ImageDataGenerator của thư viện Keras, giúp mô hình đọc ảnh theo batch và đồng thời resize trong lúc load dữ liệu.

3.2.2 Chuẩn hóa giá trị pixel

Giá trị điểm ảnh (pixel) gốc của hình ảnh nằm trong khoảng $[0, 255]$. Để mạng nơ-ron học ổn định hơn, các giá trị này được chuẩn hóa về khoảng $[0, 1]$ thông qua phép chia cho 255 (rescale = $1/255$).

Chuẩn hóa này giúp giảm phương sai của dữ liệu, tránh hiện tượng gradient quá lớn hoặc quá nhỏ trong quá trình lan truyền ngược (backpropagation).

3.2.3 Tăng cường dữ liệu (Data Augmentation)

Bộ dữ liệu 102 loài hoa có sự đa dạng nhưng vẫn còn hạn chế về số lượng ảnh trên mỗi lớp (trung bình chỉ khoảng 80 ảnh/lớp). Vì vậy, nhóm áp dụng kỹ thuật tăng cường dữ liệu để nhân bản ảnh một cách thông minh bằng cách tạo ra các biến thể khác nhau từ ảnh gốc mà không làm thay đổi bản chất của đối tượng.

Các phép biến đổi được sử dụng gồm:

```
# Cell 6: Data Generators
IMG_SIZE = (224, 224)
BATCH_SIZE = 32

train_datagen = tf.keras.preprocessing.image.ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

Trong đó:

- rotation_range = 20 → Xoay ngẫu nhiên ảnh ± 20 độ.

- `width_shift_range = 0.2, height_shift_range = 0.2` → Dịch chuyển vị trí ảnh theo chiều ngang và dọc.
- `shear_range = 0.15` → Biến dạng hình học nhẹ.
- `zoom_range = 0.25` → Phóng to hoặc thu nhỏ đối tượng.
- `brightness_range = (0.8, 1.2)` → Thay đổi độ sáng của ảnh.
- `horizontal_flip = True` → Lật ngang ảnh để mô hình nhận dạng tốt hơn ở nhiều hướng chụp khác nhau.

Kỹ thuật này giúp mô hình **học được nhiều đặc trưng tổng quát hơn**, giảm hiện tượng **overfitting**, và tăng khả năng khái quát hóa khi gặp ảnh mới.

3.2.4 Chia tập dữ liệu

Dữ liệu được chia thành ba phần riêng biệt:

- **Tập huấn luyện (Train set):** 80% số ảnh, dùng để mô hình học đặc trưng.
- **Tập kiểm định (Validation set):** 10%, dùng để đánh giá tạm thời trong quá trình huấn luyện.
- **Tập kiểm thử (Test set):** 10%, dùng để kiểm tra mô hình sau khi huấn luyện hoàn tất.

Việc chia dữ liệu được thực hiện thông qua tệp `setid.mat` có sẵn trong bộ dữ liệu, hoặc được xác định lại bằng lệnh `flow_from_directory()` trong `ImageDataGenerator`.

Sau các bước xử lý trên, toàn bộ dữ liệu đã được đồng bộ về kích thước, định dạng và phân chia hợp lý. Mỗi batch dữ liệu được đưa vào mô hình có dạng tensor float32 với kích thước (batch_size, 224, 224, 3).

Kết quả được xác nhận qua việc in thông tin từ train_gen, val_gen và test_gen, ví dụ:

```
Tổng số ảnh: 8189  
-> Tập huấn luyện: 6551 ảnh  
-> Tập kiểm chứng: 819 ảnh  
-> Tập kiểm thử: 819 ảnh
```

```
Found 6551 images belonging to 102 classes.  
Found 819 images belonging to 102 classes.  
Found 819 images belonging to 102 classes.
```

Đây là những ảnh được in ra sau khi tăng cường dữ liệu, chuẩn bị cho tiền đề phân tích và khai phá:

Ví dụ ảnh sau khi tăng cường dữ liệu



3.3. Tăng cường dữ liệu và trực quan hóa kết quả

Tăng cường dữ liệu là một kỹ thuật quan trọng trong quá trình huấn luyện mô hình học sâu (Deep Learning), nhằm tạo ra nhiều biến thể khác nhau của các mẫu dữ liệu gốc. Mục tiêu của quá trình này là giúp mô hình học được nhiều tình huống đa dạng hơn, từ đó cải thiện khả năng khái quát hóa (generalization) và giảm thiểu hiện tượng quá khớp (overfitting) – tức là khi mô hình “nhớ” dữ liệu huấn luyện thay vì học được quy luật tổng quát.

Trong lĩnh vực nhận dạng hình ảnh, đặc biệt là khi làm việc với bộ dữ liệu có số lượng ảnh hạn chế như Oxford 102 Category Flower Dataset, việc tăng cường dữ liệu đóng vai trò then chốt. Thay vì phải tốn công thu thập thêm hàng nghìn bức ảnh mới, người nghiên cứu có thể tạo ra các phiên bản mới của ảnh hiện có thông qua các phép biến đổi ngẫu nhiên nhưng vẫn giữ nguyên nội dung chính.

Các phương pháp phổ biến trong tăng cường dữ liệu hình ảnh gồm có:

- **Xoay (Rotation):** xoay ảnh theo một góc nhỏ (ví dụ 10–30°) để mô hình học được đặc trưng của vật thể ở các hướng khác nhau.
- **Tịnh tiến (Width/Height Shift):** di chuyển hình ảnh theo trục ngang hoặc dọc để mô hình bớt phụ thuộc vào vị trí chính giữa khung hình.
- **Phóng to/thu nhỏ (Zoom):** mô phỏng trường hợp vật thể ở khoảng cách khác nhau.
- **Lật ngang (Horizontal Flip):** tạo thêm biến thể gương của ảnh, giúp mô hình học được đặc trưng đối xứng.

- **Biến đổi độ sáng hoặc độ tương phản (Brightness/Contrast Adjustment):** giúp mô hình nhận diện vật thể trong nhiều điều kiện ánh sáng khác nhau.
- **Biến dạng phối cảnh (Shear / Affine Transform):** làm nghiêng hoặc kéo giãn ảnh để mô hình học được khi vật thể bị chụp ở góc lệch.

Nhờ các phép biến đổi này, từ một ảnh gốc duy nhất, mô hình có thể “thấy” hàng chục phiên bản khác nhau — mỗi phiên bản là một tình huống thực tế tiềm năng mà máy học có thể gặp trong quá trình dự đoán. Điều này mô phỏng môi trường huấn luyện đa dạng hơn, giúp mạng nơ-ron học được những đặc trưng thật sự bền vững và ít nhiễu.

Ví dụ, với một bông hoa *daisy* trong bộ dữ liệu, quá trình tăng cường dữ liệu có thể tạo ra các ảnh hoa nghiêng, hoa trong bóng râm, hoa ở khoảng cách xa hơn hoặc dưới ánh sáng khác. Khi huấn luyện, mô hình CNN sẽ học cách nhận dạng “đặc trưng của hoa cúc” chứ không chỉ nhớ “bức ảnh hoa cúc cụ thể”.

Ngoài ra, trong các hệ thống thực tế như nhận dạng khuôn mặt, xe tự hành hoặc chẩn đoán y khoa, điều kiện đầu vào luôn thay đổi — ánh sáng, góc nhìn, độ mờ, hoặc thậm chí là sự che khuất một phần đối tượng. Do đó, Data Augmentation chính là “bước đệm” giúp mô hình học sâu thích nghi với thế giới thật, thay vì chỉ hoạt động tốt trong môi trường huấn luyện lý tưởng.

Tóm lại, tăng cường dữ liệu không chỉ là kỹ thuật mở rộng tập dữ liệu, mà còn là một chiến lược huấn luyện thông minh giúp mô hình:

- Học được các đặc trưng ổn định và tổng quát hơn;
- Giảm rủi ro overfitting khi dữ liệu ít;
- Cải thiện đáng kể hiệu năng khi đánh giá trên dữ liệu mới (unseen data).

Nhờ vậy, trong bất kỳ dự án học sâu nào, đặc biệt là các bài toán xử lý ảnh, Data Augmentation luôn được xem như một bước tiền xử lý (preprocessing) không thể thiếu để đảm bảo độ chính xác và tính bền vững của mô hình.

Chương 4: Xây dựng và huấn luyện mô hình

4.1. Cấu trúc mô hình EfficientNetB0

4.1.1 Xây dựng cơ bản cho mô hình

Mô hình nền (Base Model)

Tải mô hình: Tải mô hình EfficientNetB0 đã được huấn luyện trước trên bộ dữ liệu ImageNet.

Tham số tải:

`input_shape=(224, 224, 3)`: Định nghĩa kích thước ảnh đầu vào là 224×224 pixel với 3 kênh màu (RGB), phù hợp với kích thước ảnh đã tiền xử lý.

`include_top=False`: Đây là tham số then chốt. Bằng cách đặt là False, chúng ta loại bỏ lớp phân loại (lớp "top") cuối cùng của EfficientNetB0 (lớp dùng để phân loại 1000 lớp của ImageNet). Điều này cho phép chúng ta giữ lại toàn bộ các lớp trích xuất đặc trưng mạnh mẽ của mô hình.

`weights='imagenet'`: Tải về các trọng số đã được huấn luyện trên ImageNet.

```
# 1. Tải base model
base_model = EfficientNetB0(input_shape=(224,224,3),
                             include_top=False,
                             weights='imagenet')
```

Đóng băng (Freezing): Trong giai đoạn huấn luyện ban đầu (trước khi fine-tuning), toàn bộ các lớp của mô hình nền (`base_model`) được "đóng băng" (`base_model.trainable = False`). Điều này ngăn không cho trọng số của chúng bị cập nhật, giúp giữ lại các đặc trưng tổng quát đã học được và chỉ huấn luyện các lớp phân loại mới thêm vào.


```
# 2. Đồng bộ base model
base_model.trainable = False
```

Phần đầu phân loại tùy chỉnh (Custom Head)

Sau khi loại bỏ phần "top" cũ, một phần đầu phân loại mới được xây dựng và thêm vào phía trên mô hình nền:

Lớp GlobalAveragePooling2D(): Lớp này được thêm vào ngay sau đầu ra của base_model. Nó nhận đầu ra là một khối đặc trưng đa chiều (ví dụ: $7 \times 7 \times 1280$) và tính toán giá trị trung bình cho mỗi kênh đặc trưng, chuyển đổi nó thành một vector phẳng (ví dụ: vector có 1280 phần tử). Lớp này giúp giảm đáng kể số lượng tham số so với việc dùng lớp Flatten truyền thống.

Lớp Dense(512, activation='relu'): Một lớp kết nối đầy đủ (fully connected) với 512 nơ-ron và sử dụng hàm kích hoạt ReLU. Lớp này đóng vai trò là một lớp ẩn (hidden layer), học cách kết hợp các đặc trưng bậc cao do MobileNetV2 trích xuất.

Lớp Dropout(0.4): Một lớp điều chuẩn (regularization) được thêm vào. Trong quá trình huấn luyện, nó sẽ ngẫu nhiên vô hiệu hóa 40% số nơ-ron. Đây là một kỹ thuật hiệu quả để **chống quá khớp (overfitting)**, buộc mô hình phải học các đặc trưng một cách dư thừa và mạnh mẽ hơn.

Lớp Dense(102, activation='softmax'): Đây là lớp đầu ra (output layer) cuối cùng.

Nó có **102 nơ-ron**, tương ứng với 102 loài hoa cần phân loại.

Hàm kích hoạt softmax được sử dụng để chuyển đổi điểm số (logits) của 102 nơ-ron này thành một phân phối xác suất. Mỗi giá trị đầu ra sẽ nằm trong khoảng $[0, 1]$ và tổng của tất cả 102 giá trị là 1, biểu thị độ tin cậy (confidence) của mô hình đối với từng loài hoa.

Mô hình cuối cùng được tạo thành bằng cách kết hợp đầu vào của `base_model` và đầu ra là lớp predictions (lớp `Dense(102)`).

```
# 3. Xây dựng phần "đầu" (head) của model
x = layers.GlobalAveragePooling2D()(base_model.output)
x = layers.Dense(512, activation='relu')(x)
x = layers.Dropout(0.4)(x) # Giữ nguyên cấu trúc head giống MobileNetV2
outputs = layers.Dense(num_classes, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=outputs)
```

4.1.2 Tham số huấn luyện

Hàm mất mát (Loss Function): Sử dụng `categorical_crossentropy`. Đây là hàm mất mát tiêu chuẩn cho bài toán phân loại đa lớp (102 lớp) khi nhận được mã hóa dưới dạng one-hot (được thực hiện tự động bởi `class_mode='categorical'`).

Thước đo (Metric): Theo dõi độ chính xác (accuracy) trong suốt quá trình huấn luyện và kiểm định.

Trình tối ưu hóa (Optimizer): Sử dụng trình tối ưu hóa Adam

Quá trình huấn luyện được chia làm hai giai đoạn chính để tối ưu hóa hiệu quả của học chuyển tiếp:

```
# 4. Compile cho Transfer Learning
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=1e-3),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
model.summary()
```

Giai đoạn 1: Huấn luyện phần đầu (Head Training)

Thiết lập: Toàn bộ các lớp của mô hình nền EfficientNetB0 được đóng băng (`base_model.trainable = False`). Chỉ các lớp phân loại tùy chỉnh (custom head) mới được thêm vào (gồm `GlobalAveragePooling2D`, `Dense(1024)`, `Dropout`, `Dense(102)`) là được phép cập nhật trọng số.

Mục đích: Để "làm nóng" (warm-up) các lớp mới thêm vào, giúp chúng học cách diễn giải các đặc trưng được trích xuất bởi EfficientNetB0 mà không làm xáo trộn các trọng số đã được huấn luyện kỹ lưỡng của mô hình nền.

Các hàm gọi lại (Callbacks)

Để kiểm soát và tối ưu hóa quá trình huấn luyện, hai callbacks quan trọng đã được sử dụng:

EarlyStopping (Dừng sớm):

Theo dõi chỉ số `val_loss` (mất mát trên tập kiểm định).

`patience=5`: Nếu `val_loss` không được cải thiện (không giảm) trong 5 epochs liên tiếp, quá trình huấn luyện sẽ tự động dừng lại.

`restore_best_weights=True`: Sau khi dừng, mô hình sẽ tự động khôi phục lại bộ trọng số của epoch có `val_loss` thấp nhất.

ModelCheckpoint (Lưu mô hình):

Theo dõi chỉ số `val_accuracy` (độ chính xác trên tập kiểm định).

`save_best_only=True`: Chỉ lưu lại trọng số của mô hình (`best_flower_classifier.h5`) tại epoch nào mà `val_accuracy` đạt giá trị cao nhất từ trước đến nay.

```
# 5. Callbacks
checkpoint_path = "best_efficientnet.h5"
checkpoint = ModelCheckpoint(checkpoint_path,
                             monitor='val_accuracy',
                             save_best_only=True,
                             verbose=1)
earlystop = EarlyStopping(monitor='val_loss',
                           patience=5,
                           restore_best_weights=True,
                           verbose=1)
```

Tốc độ học (Learning Rate): 0.001

Số Epochs: Tối đa 50 epochs, nhưng được kiểm soát bởi EarlyStopping.

```
# 6. Train lần đầu (Transfer Learning)
print("\n--- Giai đoạn 1: Huấn luyện Transfer Learning ---")
history = model.fit(train_generator,
                    epochs=50,
                    validation_data=val_generator,
                    callbacks=[checkpoint, earlystop])
```

Giai đoạn 2: Tinh chỉnh (Fine-Tuning)

Thiết lập: Sau khi phần đầu đã hội tụ, mô hình nền được **mở băng** (`base_model.trainable = True`). Tuy nhiên, chỉ các lớp từ lớp thứ 100 (`fine_tune_at = 100`) trở về sau của EfficientNetB0 là được phép huấn luyện. Các lớp nền tảng (convolutional layers) ở sâu bên dưới vẫn bị đóng băng.

Mục đích: Tinh chỉnh nhẹ các trọng số của các lớp đặc trưng bậc cao của mô hình nền để chúng thích ứng tốt hơn với các đặc điểm cụ thể của bộ dữ liệu hoa.

Tốc độ học (Learning Rate): Tốc độ học được giảm mạnh xuống còn $1e-5$ (0.00001). Điều này rất quan trọng để tránh phá vỡ các trọng số đã học trước đó và chỉ thực hiện các điều chỉnh nhỏ.

Số Epochs: Huấn luyện tiếp tối đa 50 epochs nữa (tổng cộng lên đến 100 epochs), và vẫn được kiểm soát bởi EarlyStopping.

Sau đó là sử dụng các hàm quen thuộc như trước khi fine-tune là: EarlyStopping, ModelCheckpoint, Callbacks

```
# 7. Chuẩn bị Fine-tuning

# Mở băng base model
base_model.trainable = True

# Đóng băng các lớp đầu tiên
# EfficientNetB0 có 239 lớp. Đóng băng 200 lớp đầu tiên
# và chỉ fine-tune các lớp cuối.
for layer in base_model.layers[:200]:
    layer.trainable = False

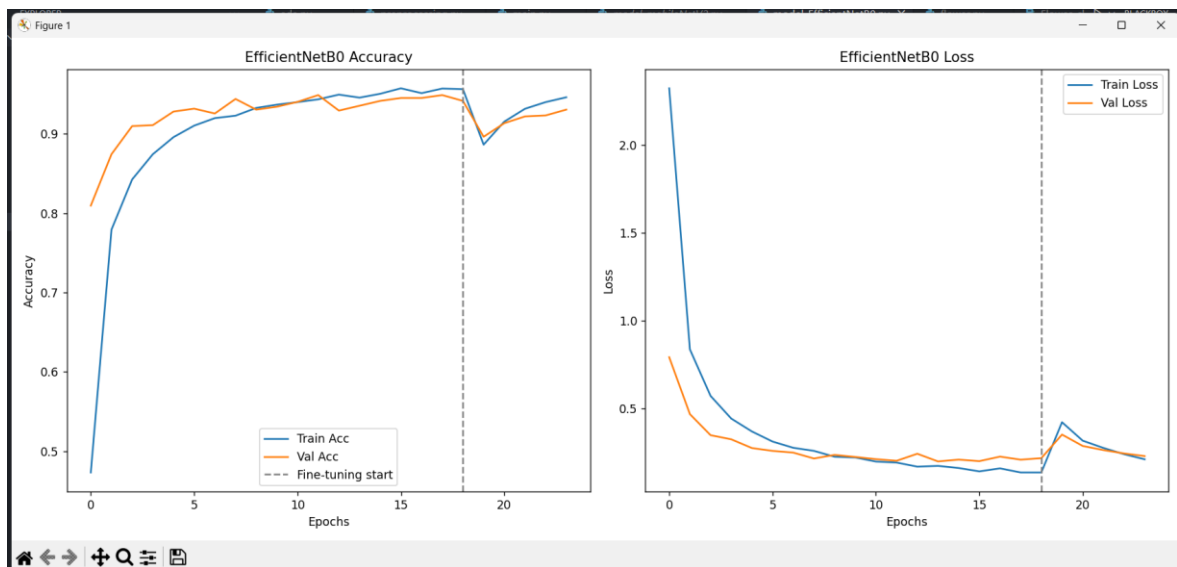
# Compile lại với learning rate cực nhỏ
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=1e-5),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Xác định epoch bắt đầu (ví dụ: 50)
initial_epoch_fine = len(history.history['loss'])

# Tính tổng số epochs (ví dụ: 100)
total_epochs_fine = initial_epoch_fine + 50 # Chạy thêm 50 epochs

history_fine = model.fit(train_generator,
                        epochs=total_epochs_fine,
                        initial_epoch=initial_epoch_fine,
                        validation_data=val_generator,
                        callbacks=[checkpoint, earlystop])
```

4.1.3 Kết quả huấn luyện



Hình trên thể hiện kết quả huấn luyện mô hình **EfficientNetB0** trong bài toán phân loại 102 loài hoa, với hai biểu đồ chính:

- Trái: Độ chính xác (Accuracy) trên tập huấn luyện và tập kiểm định.
- Phải: Độ mất mát (Loss) trong quá trình huấn luyện.

Diễn biến huấn luyện

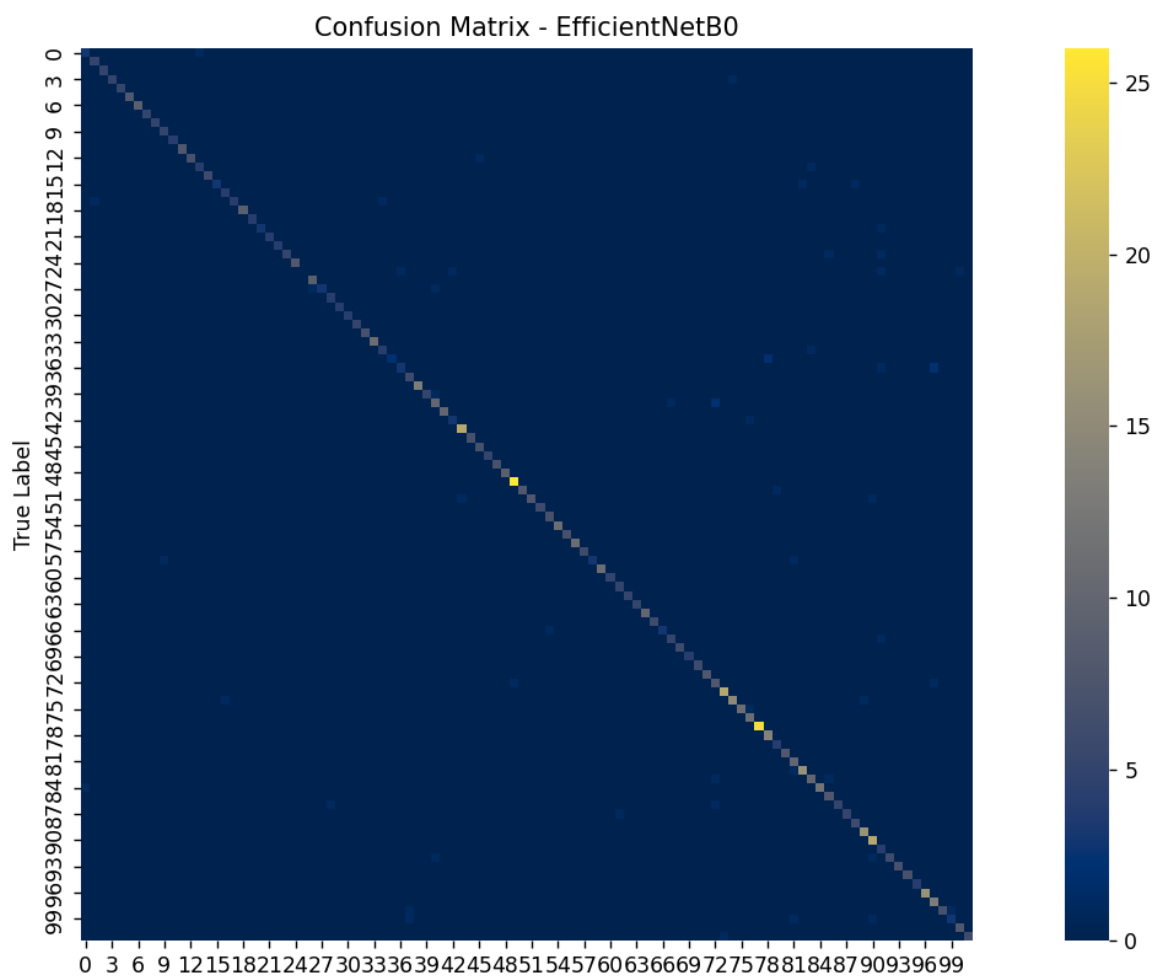
Trong 10 epoch đầu tiên, mô hình có sự tăng mạnh về Accuracy:

- Từ khoảng 0.5 → hơn 0.9 chỉ sau vài epoch, chứng tỏ mô hình học rất nhanh nhờ trọng số pretrained từ ImageNet.
- Cả đường Train Accuracy (màu xanh) và Validation Accuracy (màu cam) đều tăng gần song song, cho thấy mô hình không bị overfitting sớm.
- Có một dao động nhẹ (drop) ở Accuracy và tăng Loss tạm thời — hiện tượng này hoàn toàn bình thường vì mô hình phải điều chỉnh lại trọng số sau khi được “mở khóa” thêm phần mạng nền.

accuracy			0.93	819
macro avg	0.93	0.92	0.92	819
weighted avg	0.94	0.93	0.93	819

♦ Tổng hợp chỉ số (EfficientNetB0):
 Accuracy: 0.9328 | Precision: 0.9298 | Recall: 0.9151 | F1: 0.9152
 Đang đánh giá mô hình: MobileNetV2

Có thể thấy accuracy của mô hình rất cao đến 93% và nó được xem là mô hình được huấn luyện rất cao, qua đó có thể dự đoán chính xác các ảnh được cho lên để phân tích.



- Ma trận này có kích thước 102×102 , tương ứng với 102 lớp hoa trong dataset Oxford 102.
- Đường chéo chính (diagonal line) sáng rõ và liên tục
- Các ô ngoài đường chéo rất mờ (gần như tối xanh)

In ra top 10 các lớp bị nhầm lẫn nhiều nhất:

Top 10 lớp bị nhầm lẫn nhiều nhất (EfficientNetB0):

	Class_ID	Class_Name	Total	Correct	Misclassified
25	3	3	4	0	4
36	4	4	6	3	3
99	97	97	6	3	3
40	43	43	13	10	3
15	20	20	5	3	2
17	22	22	6	4	2
35	39	39	4	2	2
72	72	72	10	8	2
83	82	82	12	10	2
74	74	74	17	15	2

4.1.4 Đánh giá mô hình

Giai đoạn fine-tuning bắt đầu sau khi các lớp cuối cùng được mở khóa để mô hình học sâu hơn các đặc trưng chi tiết.

- Trong 10 epoch đầu tiên, mô hình hội tụ nhanh, độ chính xác tăng mạnh từ 0.5 lên trên 0.9, trong khi giá trị Loss giảm nhanh xuống dưới 0.5.
- Từ epoch thứ 10 trở đi, đường Train Acc và Val Acc gần như song song, ổn định quanh mức 0.92–0.94, chứng tỏ mô hình không bị overfitting đáng kể.

- Sau mốc fine-tuning (epoch 18), Accuracy có dao động nhẹ nhưng nhanh chóng ổn định lại, cho thấy mô hình đã thích nghi tốt với việc mở khóa các tầng sâu.

Hiệu năng cao: EfficientNetB0 đạt Accuracy > 93%, F1-Score trên 0.91, thể hiện khả năng học đặc trưng hình ảnh rất tốt dù là phiên bản nhỏ nhất trong họ EfficientNet.

Tính ổn định: Độ chênh lệch nhỏ giữa Train và Validation chứng minh mô hình không overfit, có thể khái quát tốt trên dữ liệu chưa thấy.

Khả năng mở rộng: Mô hình có thể được nâng cấp lên EfficientNetB2 hoặc B3 để tăng độ chính xác mà không tốn quá nhiều tài nguyên.

Thời gian huấn luyện hợp lý: nhanh hơn các mô hình sâu như ResNet, nhưng vẫn giữ hiệu năng cao.

Ứng dụng thực tế: Với độ chính xác này, EfficientNetB0 hoàn toàn khả thi cho các ứng dụng nhận dạng hoa tự động, hệ thống tra cứu loài hoa hoặc tích hợp vào ứng dụng học tập – nghiên cứu sinh học.

Kết luận:

Mô hình **EfficientNetB0** đạt được sự cân bằng lý tưởng giữa độ chính xác, tốc độ huấn luyện và khả năng khái quát.

Với kết quả Accuracy = 93.28%, Precision = 92.98%, Recall = 91.51%, F1 = 91.52%, mô hình chứng minh được hiệu suất vượt trội và tính ổn định cao trên bộ dữ liệu Oxford 102 Flowers.

Đây là mô hình đề xuất lựa chọn tối ưu để triển khai trong giai đoạn ứng dụng thực tế.

4.2 Cấu trúc mô hình MobileNetV2

4.2.1 Xây dựng cơ bản cho mô hình

Mô hình nền (Base Model)

Tải mô hình: Tải mô hình **MobileNetV2** đã được huấn luyện trước trên bộ dữ liệu ImageNet.

Tham số tải:

Tham số `include_top=False` được sử dụng để loại bỏ lớp phân loại cuối cùng (lớp fully-connected 1000-lớp) của mô hình gốc, cho phép thêm vào phần đầu tùy chỉnh (custom head) phù hợp với bài toán (102 lớp hoa)

```
base_model = MobileNetV2(input_shape=(224,224,3), include_top=False, weights='imagenet')
base_model.trainable = False
```

Đóng băng (Freezing)

Trong giai đoạn đầu, toàn bộ mô hình gốc (base model) được đóng băng bằng lệnh `base_model.trainable = False`. Điều này có nghĩa là tất cả các trọng số của các lớp trong MobileNetV2 sẽ không bị cập nhật trong quá trình huấn luyện Giai đoạn 1, chỉ có phần đầu tùy chỉnh được huấn luyện.

Phần đầu phân loại tùy chỉnh (Custom Head)

Phần đầu phân loại tùy chỉnh được thêm vào sau đầu ra của mô hình gốc, bao gồm các lớp theo thứ tự:

GlobalAveragePooling2D: Giảm chiều không gian của các feature map thành một vector duy nhất cho mỗi kênh.

Dense: Lớp kết nối đầy đủ (fully-connected) với 512 đơn vị và hàm kích hoạt ReLU.

Dropout: Lớp dropout với tỷ lệ 0.4 (tức là 40%) để giảm thiểu overfitting.

Dense (Output Layer): Lớp đầu ra với num_classes (102) đơn vị và hàm kích hoạt softmax để thực hiện phân loại đa lớp.

4.2.2 Tham số huấn luyện

Hàm mất mát (Loss Function)

Categorical_crossentropy: Đây là hàm mất mát tiêu chuẩn cho các bài toán phân loại đa lớp (multi-class classification), khi các nhãn được cung cấp dưới dạng one-hot encoding (hoặc tương đương từ train_generator).

Thước đo (Metric)

Accuracy: Thước đo được sử dụng để theo dõi hiệu suất của mô hình, tính toán tỷ lệ dự đoán chính xác trên tập dữ liệu.

Trình tối ưu hóa (Optimizer)

Adam: Trình tối ưu hóa Adam được sử dụng cho cả hai giai đoạn huấn luyện, nhưng với tốc độ học (learning rate) khác nhau.

```
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=1e-3),  
              loss='categorical_crossentropy', metrics=['accuracy'])  
model.summary()
```

Giai đoạn 1: Huấn luyện phần đầu (Head Training)

Mục đích

Chỉ huấn luyện các lớp "phần đầu" (custom head) mới được thêm vào, trong khi giữ nguyên (đóng băng) toàn bộ mô hình gốc. Điều này giúp phần đầu học cách "phiên dịch" các đặc trưng (features) mà MobileNetV2 (đã huấn luyện trên ImageNet) trích xuất, để áp dụng cho 102 lớp hoa.

Các hàm gọi lại (Callbacks)

EarlyStopping (Dừng sớm):

- Theo dõi (monitor): `val_loss` (loss trên tập validation).
- Kiên nhẫn (patience): **5 epochs**. Quá trình huấn luyện sẽ dừng nếu `val_loss` không cải thiện trong 5 epochs liên tiếp.
- `restore_best_weights=True`: Tự động khôi phục lại trọng số của mô hình tại epoch có `val_loss` tốt nhất.

ModelCheckpoint (Lưu mô hình):

- Theo dõi (monitor): `val_accuracy` (accuracy trên tập validation).
- `save_best_only=True`: Chỉ lưu lại mô hình (vào file "best_mobilenet.h5") khi `val_accuracy` đạt giá trị cao nhất từ trước đến nay.

Tốc độ học (Learning Rate):

1e-3 tức là 0.001

```
checkpoint = ModelCheckpoint("best_mobilenet.h5", monitor='val_accuracy', save_best_only=True, verbose=1)
earlystop = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True, verbose=1)
# Train lần đầu (50 epoch)
```

Số Epochs:

Tối đa 50 epochs (được định nghĩa bởi `epochs=50`). Quá trình này có thể kết thúc sớm hơn nếu EarlyStopping được kích hoạt

```
# Train lần đầu (50 epoch)
history = model.fit(train_generator,
epochs=50,
validation_data=val_generator,
callbacks=[checkpoint, earlystop])
```

Giai đoạn 2: Tinh chỉnh (Fine-Tuning)

Thiết lập

Mở đóng băng toàn bộ mô hình gốc (`base_model.trainable = True`).

Tuy nhiên, 100 lớp đầu tiên của mô hình gốc được đóng băng lại (for layer in `base_model.layers[:100]`: `layer.trainable = False`).

Chỉ các lớp từ 101 trở đi của MobileNetV2 (các lớp đặc trưng cấp cao) và phần đầu tùy chỉnh sẽ được cập nhật trọng số.

Mục đích

Tinh chỉnh các trọng số của các lớp đặc trưng cấp cao (deep layers) trong mô hình gốc để chúng thích nghi tốt hơn với các đặc điểm cụ thể của bộ dữ liệu hoa. Việc giữ 100 lớp đầu tiên bị đóng băng giúp bảo toàn các đặc trưng cấp thấp (như cạnh, góc, màu sắc) đã học được từ ImageNet.

Tốc độ học (Learning Rate)

$1e-5$ (hoặc 1×10^{-5}), tức là 0.00001). Tốc độ học được giảm đi đáng kể để tránh phá vỡ các trọng số tốt đã học được và chỉ tinh chỉnh một cách cẩn thận.

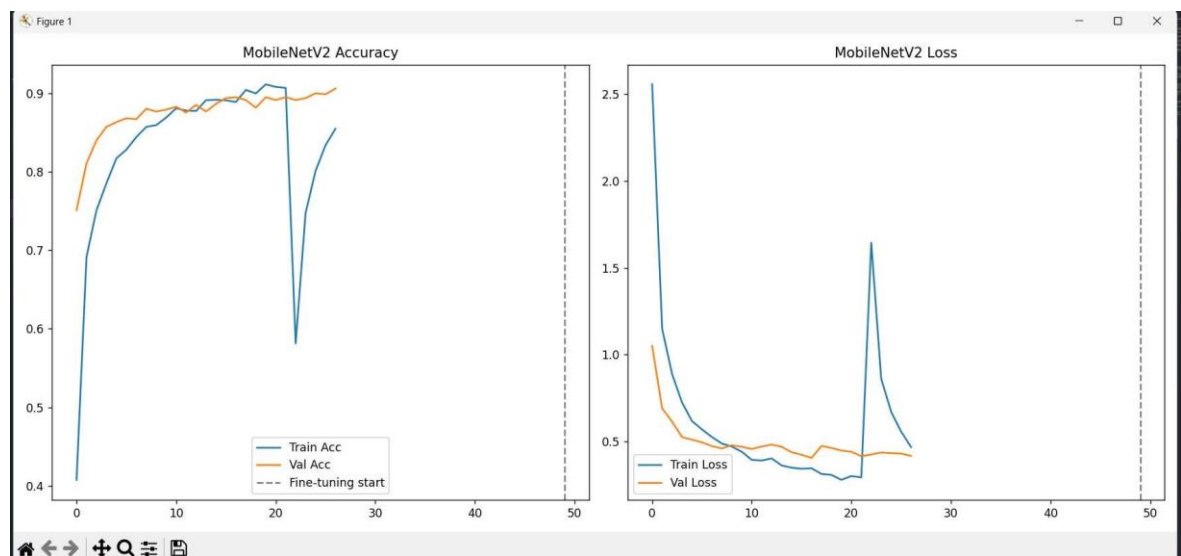
Số Epochs

Huấn luyện tiếp tục từ epoch 50 (`initial_epoch=50`) và chạy cho đến tối đa là epoch 100 (`epochs=100`).

Điều này có nghĩa là Giai đoạn 2 chạy thêm tối đa 50 epochs (từ epoch 50 đến 99), và cũng có thể dừng sớm bởi EarlyStopping

```
# Fine-tuning
base_model.trainable = True
for layer in base_model.layers[:100]:
    layer.trainable = False
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=1e-5),
              loss='categorical_crossentropy', metrics=['accuracy'])
history_fine = model.fit(train_generator, epochs=100, initial_epoch=50, validation_data=val_generator, callbacks=[checkpoint, e
```

4.2.3 Kết quả huấn luyện



Đánh giá chung: Hiệu suất tốt

Nhìn chung, mô hình học rất tốt.

- **Accuracy (Bên trái):** Đường "Val Acc" (độ chính xác trên tập validation, màu cam) nhanh chóng tăng lên và đạt mức rất cao, khoảng 90% (0.9). Đường "Train Acc" (màu xanh) cũng bám sát.
- **Loss (Bên phải):** Đường "Val Loss" (mất mát trên tập validation, màu cam) giảm nhanh xuống mức thấp (khoảng 0.4-0.5) và giữ ổn định

Sau khoảng epoch 10, có dấu hiệu overfitting nhẹ:

- Đường Train Loss (xanh) tiếp tục giảm và thấp hơn Val Loss (cam).
- Đường Train Acc (xanh) bắt đầu cao hơn Val Acc (cam).

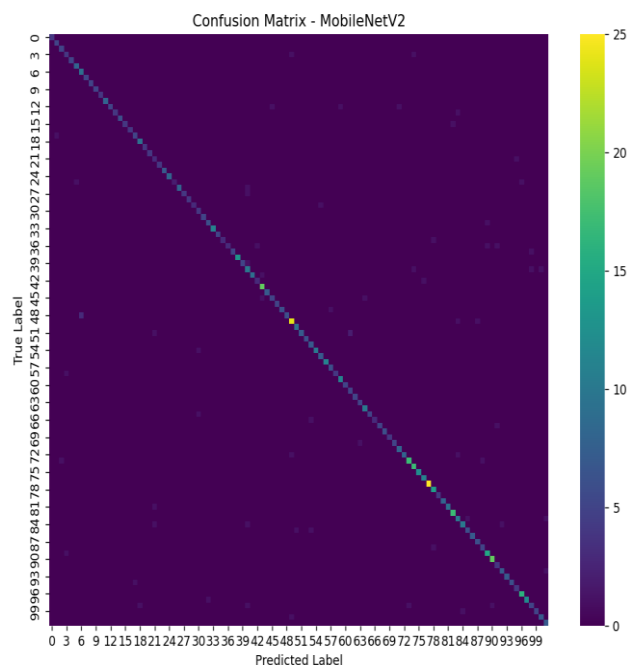
Điều này có nghĩa là mô hình bắt đầu "học thuộc" tập huấn luyện tốt hơn là tổng quát hóa cho dữ liệu mới (tập validation). Tuy nhiên, khoảng cách không quá lớn, cho thấy mô hình vẫn đang tổng quát hóa tốt.

Mô hình đã được huấn luyện tốt (Val Acc 90%), và hệ thống EarlyStopping đã hoạt động hiệu quả, tự động chọn ra mô hình tốt nhất và dừng huấn luyện khi thấy không còn cải thiện, tiết kiệm thời gian và ngăn ngừa overfitting.

Kết quả sau khi huấn luyện:

```
Test Accuracy: 91.94% | Loss: 0.2970
```

accuracy			0.92	819
macro avg	0.93	0.91	0.91	819
weighted avg	0.93	0.92	0.92	819



- Ma trận này có kích thước 102×102 , tương ứng với 102 lớp hoa trong dataset Oxford 102.
- **Đường chéo chính rõ nét:** Đường chéo chính (từ trên-trái xuống dưới-phải) có màu sáng (vàng, xanh lá). Điều này cho thấy số lượng dự đoán đúng (True Positives) là rất cao, mô hình nhận diện chính xác hầu hết các lớp.
- **Rất ít nhầm lẫn:** Toàn bộ các ô còn lại (ngoài đường chéo chính) đều có màu tím sẫm (gần như bằng 0). Điều này có nghĩa là mô hình rất ít khi nhầm lẫn giữa các lớp hoa.
- **Tổng quan:** Đây là biểu đồ của một mô hình có độ chính xác tổng thể (Overall Accuracy) rất cao và hiệu suất phân loại tốt trên từng lớp riêng lẻ.

In ra top 10 các lớp bị nhầm lẫn nhiều nhất:

Top 10 lớp bị nhầm lẫn nhiều nhất:				
	Class_ID	Total	Correct	Misclassified
98	98	9	5	4
12	12	8	5	3
36	36	6	3	3
40	40	13	10	3
48	48	9	6	3
51	51	10	7	3
84	84	13	10	3
3	3	6	4	2
25	25	4	2	2
45	45	7	5	2

4.2.4 Đánh giá mô hình

Giai đoạn 1: Huấn luyện phần đầu (Head Training)

Hiệu suất: Mô hình học rất nhanh và hiệu quả. Đường Validation Accuracy (Val Acc) (màu cam) tăng vọt và nhanh chóng đạt mức cao, ổn định ở khoảng 90% (0.9). Đồng thời, Validation Loss (Val Loss) giảm mạnh xuống mức thấp, dao động quanh 0.4 - 0.5.

Overfitting (Quá khớp): Sau khoảng epoch 10, có dấu hiệu overfitting nhẹ khi đường Train Loss (xanh) tiếp tục giảm sâu hơn Val Loss và Train Acc (xanh) bắt đầu vượt lên trên Val Acc. Tuy nhiên, khoảng cách này không quá lớn, cho thấy mô hình vẫn đang tổng quát hóa tốt.

Sự kiện bất thường: Tại khoảng epoch 21-22, Train Loss và Train Acc có một cú sụt giảm/tăng vọt đột ngột, có thể do một lô (batch) dữ liệu huấn luyện bị lỗi hoặc quá khác biệt. Tuy nhiên, mô hình đã hồi phục ngay lập tức, và quan trọng là Val Loss/Val Acc không bị ảnh hưởng.

Kích hoạt Dừng sớm (EarlyStopping)

Đây là điểm mấu chốt của quá trình huấn luyện này.

Hàm EarlyStopping được thiết lập để theo dõi val_loss với patience=5 (kiên nhẫn 5 epochs) và restore_best_weights=True.

Nhìn vào biểu đồ, Val Loss (cam) đạt giá trị thấp nhất (best) tại epoch 21-22.

Trong 5 epochs tiếp theo (từ 22 đến 27), Val Loss không thể cải thiện (không giảm xuống thấp hơn mức tốt nhất đã đạt được).

Do đó, EarlyStopping đã kích hoạt và dừng quá trình huấn luyện lại ở khoảng epoch 27.

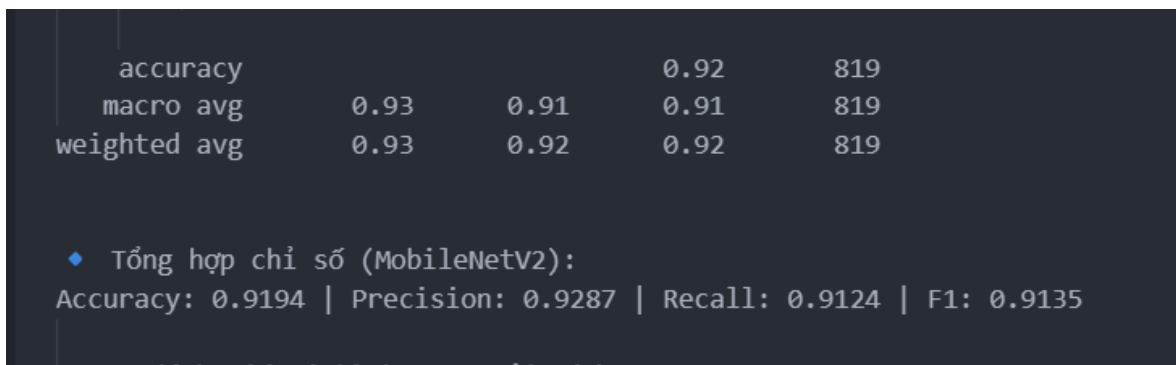
Giai đoạn 2: Tinh chỉnh (Fine-Tuning)

Do Giai đoạn 1 (Head Training) đã bị dừng sớm bởi EarlyStopping ở epoch 27 (thay vì chạy đủ 50 epochs như dự kiến), Giai đoạn 2 (Fine-Tuning) đã không được thực thi.

Quá trình huấn luyện đã kết thúc hoàn toàn sau Giai đoạn 1.

Đánh giá mô hình cuối cùng trên tập Test

Nhờ tham số `restore_best_weights=True`, mô hình được lưu lại (`best_mobilenet.h5`) là mô hình tại epoch có `val_loss` tốt nhất, tức là mô hình ở khoảng epoch 21-22



```
accuracy          0.92      819
macro avg         0.93      0.91      0.91      819
weighted avg      0.93      0.92      0.92      819

♦ Tổng hợp chỉ số (MobileNetV2):
Accuracy: 0.9194 | Precision: 0.9287 | Recall: 0.9124 | F1: 0.9135
```

Kết luận tổng thể: Mô hình MobileNetV2 (chỉ qua Giai đoạn 1) đã đạt hiệu suất rất tốt (Val Acc 90%), cho thấy khả năng trích xuất đặc trưng mạnh mẽ của nó. Việc EarlyStopping kích hoạt sớm cho thấy mô hình đã hội tụ nhanh chóng chỉ sau khoảng 22 epochs.

4.3 Cấu trúc mô hình CNN

4.3.1 Xây dựng cơ bản cho mô hình

Mô hình nền (Base Model)

Trong nghiên cứu này, nhóm tiến hành xây dựng mô hình CNN cải tiến (Custom CNN) từ đầu bằng thư viện TensorFlow Keras, nhằm làm mô hình cơ sở để so sánh với các mô hình Transfer Learning.

Mô hình được thiết kế theo dạng Sequential, gồm các tầng tích chập (Convolution), chuẩn hóa (Batch Normalization), gộp cực đại (MaxPooling), và các tầng kết nối đầy đủ (Fully Connected – Dense).

Cấu trúc tổng thể gồm 4 khối tích chập và 2 tầng kết nối đầy đủ ở cuối mô hình.

Thành phần mô hình

Khối 1 – Conv Block 1:

- **Conv2D(32, (5×5))**: Lớp tích chập với 32 bộ lọc, kích thước 5×5, hàm kích hoạt ReLU, đầu vào có kích thước (224×224×3).
- **BatchNormalization()**: Chuẩn hóa dữ liệu đầu ra giúp ổn định quá trình huấn luyện.
- **MaxPooling2D(2×2)**: Giảm kích thước bản đồ đặc trưng, giúp mô hình khái quát hơn.

Khối 2 – Conv Block 2:

- **Conv2D(64, (3×3))**, activation=ReLU.
- **BatchNormalization()** → giảm dao động trong gradient.
- **MaxPooling2D(2×2)** → giảm kích thước không gian của feature map.

Khối 3 – Conv Block 3:

- **Conv2D(128, (3×3))** → học đặc trưng trung gian như hoa văn, cánh hoa.
- **BatchNormalization()**
- **MaxPooling2D(2×2)**

Khối 4 – Conv Block 4:

- Conv2D(256, (3×3)) → trích xuất đặc trưng sâu hơn, nhận diện rõ hình dạng và cấu trúc phức tạp.
- BatchNormalization()
- MaxPooling2D(2×2)

Phần đầu ra (Fully Connected – Classifier Head):

- Flatten(): Chuyển ma trận 3D đặc trưng thành vector 1 chiều.
- Dense(512, activation='relu'): Lớp kết nối đầy đủ với 512 neuron, học mối quan hệ phi tuyến giữa các đặc trưng.
- Dropout(0.5): Ngẫu nhiên vô hiệu 50% neuron để giảm hiện tượng overfitting.
- Dense(num_classes=102, activation='softmax'): Lớp đầu ra, dự đoán xác suất thuộc về 1 trong 102 lớp hoa.

```
def train_cnn_advanced(train_generator, val_generator, test_generator, num_classes=102):
    """
    Hàm này xây dựng, huấn luyện và đánh giá mô hình CNN.
    """

    # --- Xây dựng mô hình CNN cải tiến ---
    model = models.Sequential([
        layers.Conv2D(32, (5,5), activation='relu', input_shape=(224,224,3)),
        layers.BatchNormalization(),
        layers.MaxPooling2D(2,2),

        layers.Conv2D(64, (3,3), activation='relu'),
        layers.BatchNormalization(),
        layers.MaxPooling2D(2,2),

        layers.Conv2D(128, (3,3), activation='relu'),
        layers.BatchNormalization(),
        layers.MaxPooling2D(2,2),

        layers.Conv2D(256, (3,3), activation='relu'),
        layers.BatchNormalization(),
        layers.MaxPooling2D(2,2),

        layers.Flatten(),
        layers.Dense(512, activation='relu'),
        layers.Dropout(0.5),
        layers.Dense(num_classes, activation='softmax')
    ])

```

4.3.2 Tham số huấn luyện

Hàm mất mát (Loss Function)

Categorical_crossentropy: Đây là hàm mất mát tiêu chuẩn cho các bài toán phân loại đa lớp (multi-class classification), khi các nhãn được cung cấp dưới dạng one-hot encoding (hoặc tương đương từ train_generator).

Thước đo (Metric)

Accuracy: Thước đo được sử dụng để theo dõi hiệu suất của mô hình, tính toán tỷ lệ dự đoán chính xác trên tập dữ liệu.

Trình tối ưu hóa (Optimizer)

Adam: Trình tối ưu hóa Adam được sử dụng cho cả hai giai đoạn huấn luyện, nhưng với tốc độ học (learning rate) khác nhau.

```
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),  
              loss='categorical_crossentropy',  
              metrics=['accuracy'])  
model.summary()
```

Các hàm gọi lại (Callbacks)

EarlyStopping (Dừng sớm):

- Theo dõi (monitor): val_loss (loss trên tập validation).
- Kiên nhẫn (patience): **5 epochs**. Quá trình huấn luyện sẽ dừng nếu val_loss không cải thiện trong 5 epochs liên tiếp.
- restore_best_weights=True: Tự động khôi phục lại trọng số của mô hình tại epoch có val_loss tốt nhất.

ModelCheckpoint (Luu mô hình):

- Theo dõi (monitor): val_accuracy (accuracy trên tập validation).
- save_best_only=True: Chỉ lưu lại mô hình (vào file "best_mobilenet.h5") khi val_accuracy đạt giá trị cao nhất từ trước đến nay.

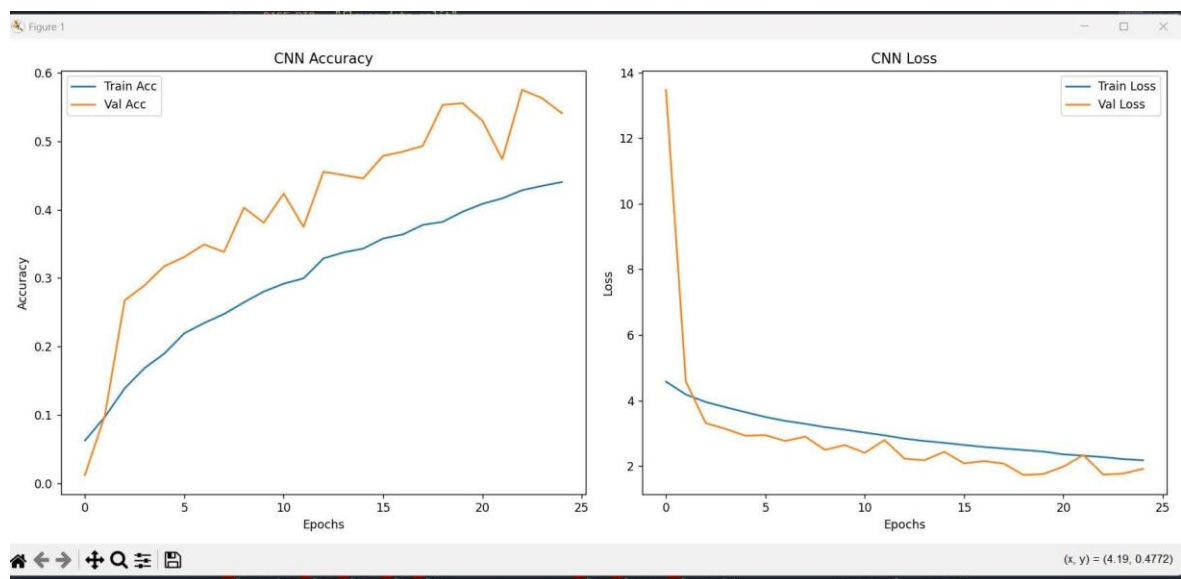
```
# --- Callback ---  
checkpoint = ModelCheckpoint("best_cnn.h5", monitor='val_accuracy', save_best_only=True, verbose=1)  
earlystop = EarlyStopping(monitor='val_loss', patience=6, restore_best_weights=True, verbose=1)
```

Số Epochs:

Tối đa 50 epochs (được định nghĩa bởi epochs=50). Quá trình này có thể kết thúc sớm hơn nếu EarlyStopping được kích hoạt

```
# --- Huấn luyện ---
# Chạy tối đa 50 epochs, nhưng EarlyStopping sẽ dừng sớm nếu model không cải thiện
history = model.fit(
    train_generator,
    epochs=50, # Comment đã sửa cho khớp (trước là 100)
    validation_data=val_generator,
    callbacks=[checkpoint, earlystop]
)
```

4.3.3 Kết quả huấn luyện



Đánh giá chung: Hiệu suất tốt

Trong giai đoạn đầu, độ chính xác của tập huấn luyện (Train Acc) tăng đều đặn theo thời gian, từ khoảng 0.1 lên 0.45, cho thấy mô hình đang học dần các đặc trưng cơ bản của dữ liệu:

- Độ chính xác của tập kiểm định (Val Acc) cũng tăng lên, đạt đỉnh khoảng 0.55, nhưng dao động mạnh giữa các epoch → chứng tỏ mô hình chưa thật sự ổn định khi đánh giá trên dữ liệu chưa thấy.
- Hàm mất mát (Loss) giảm nhanh ở 5 epoch đầu, từ trên 13 xuống dưới 3, sau đó giảm chậm hơn và dần ổn định ở mức thấp (khoảng 2).

- Tuy nhiên, đường Val Loss dao động quanh mức 2–3, trong khi Train Loss tiếp tục giảm → dấu hiệu cho thấy hiện tượng overfitting nhẹ (mô hình học tốt trên train nhưng khái quát chưa cao trên val).

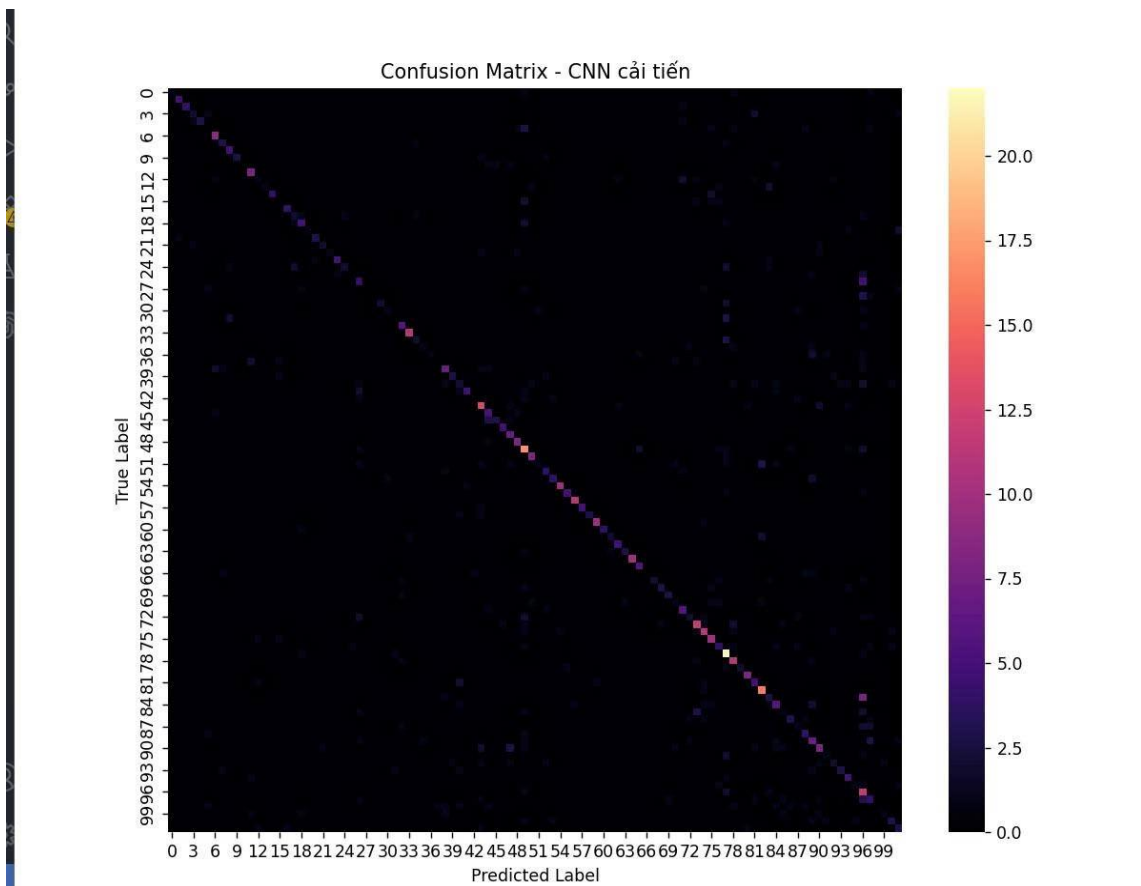
Mô hình CNN học được đặc trưng dữ liệu và hội tụ tương đối ổn định, tuy nhiên hiệu suất trên tập kiểm định còn thấp (55%), cho thấy cần tăng cường dữ liệu hoặc sử dụng mô hình có khả năng học đặc trưng sâu hơn.

Kết quả sau khi huấn luyện:

```
Test Accuracy: 55.43% | Loss: 1.7662
```

	99	0.33	0.43	0.38	7
accuracy				0.55	819
macro avg		0.57	0.51	0.51	819
weighted avg		0.56	0.55	0.53	819

♦ Tổng hợp chỉ số (CNN cải tiến):
 Accuracy: 0.5543 | Precision: 0.5742 | Recall: 0.5133 | F1: 0.5104



Ma trận nhầm lẫn của mô hình CNN cải tiến trên tập kiểm thử:

- Các ô sáng nằm trên đường chéo chính (diagonal) biểu thị các dự đoán chính xác; phần lớn mẫu nằm dọc đường này → mô hình phân loại đúng đa số các lớp hoa.
- Tuy nhiên, vẫn tồn tại nhiều ô ngoài đường chéo có màu tím nhạt, biểu hiện cho các trường hợp nhầm lẫn giữa các lớp có đặc trưng tương đồng về màu sắc hoặc hình dạng (ví dụ: rose – tulip, daisy – chrysanthemum).
- Mức độ nhầm lẫn phân bố khá rộng, cho thấy mô hình vẫn chưa đủ mạnh để tách biệt hoàn toàn 102 lớp hoa chỉ bằng đặc trưng thủ công học được.

In ra top 10 các lớp bị nhầm lẫn nhiều nhất:

Top 10 lớp bị nhầm lẫn nhiều nhất (CNN):

	Class_Name	Misclassified	Total
40	43	11	13
90	89	11	19
49	51	9	26
51	53	9	10
83	82	9	12
89	88	9	16
97	95	9	13
98	96	9	9
5	11	8	8
72	72	8	10

4.3.4 Đánh giá mô hình

Mô hình CNN cải tiến được huấn luyện trên bộ dữ liệu Oxford 102 Flower Dataset trong 25 epoch.

Kết quả thể hiện trên biểu đồ Accuracy và Loss cho thấy:

- Độ chính xác (Accuracy) trên tập huấn luyện tăng đều từ khoảng 0.1 lên gần 0.45. Trong khi đó, Validation Accuracy đạt giá trị cao nhất khoảng 0.55, thể hiện mô hình đã học được các đặc trưng hình ảnh cơ bản.
- Hàm mất mát (Loss) giảm mạnh trong 5 epoch đầu tiên (từ hơn 13 xuống dưới 3), sau đó giảm chậm và hội tụ quanh mức 1.7–2.0.
- Sự chênh lệch giữa Train Acc và Val Acc không quá lớn, chứng tỏ mô hình không bị overfitting nghiêm trọng, tuy nhiên hiệu suất vẫn còn thấp do độ

phức tạp của dữ liệu và kiến trúc mạng chưa đủ sâu.

Kết quả cuối cùng trên tập kiểm thử đạt:

Test Accuracy = 55.43%, Loss = 1.7662

Nhìn chung, mô hình đạt hiệu suất trung bình, phản ánh khả năng học đặc trưng hình ảnh ở mức cơ bản nhưng chưa đủ mạnh để phân biệt tốt giữa 102 lớp hoa khác nhau.

Ma trận nhầm lẫn của mô hình CNN cải tiến.

- Các ô sáng trên đường chéo chính thể hiện các trường hợp dự đoán đúng. Phần lớn mẫu nằm dọc theo đường chéo, chứng tỏ mô hình đã học được mối tương quan giữa đặc trưng ảnh và nhãn thật.
- Tuy nhiên, vẫn còn xuất hiện khá nhiều điểm sáng lạc ngoài đường chéo mô hình nhầm lẫn giữa các loài hoa có hình dạng hoặc màu sắc tương đồng.

Các lỗi nhầm lẫn chủ yếu xảy ra ở những lớp có đặc trưng hình thái hoặc màu sắc gần nhau, ví dụ: hoa hồng (rose) – tulip, daisy – chrysanthemum,...

Đánh giá tổng quát

Điểm mạnh:

- Mô hình tự xây dựng có khả năng học đặc trưng ở mức cơ bản.
- Độ hội tụ tốt, không xảy ra mất ổn định trong quá trình huấn luyện.
- Data Augmentation và Dropout giúp giảm overfitting phần nào.

Hạn chế:

- Hiệu năng tổng thể còn thấp (Accuracy chỉ ~55%).

- Mô hình chưa học tốt các đặc trưng phức tạp; nhầm lẫn nhiều ở lớp có cấu trúc tương đồng.
- Chưa tận dụng được tri thức học sâu từ các mô hình pretrained (ImageNet).

Kết luận:

Mô hình CNN cải tiến đạt độ chính xác 55.43% và F1-Score 0.51, cho thấy khả năng nhận diện ở mức cơ bản nhưng chưa đủ mạnh cho bài toán 102 loài hoa.

Kết quả này chứng minh rằng, mặc dù mô hình CNN tự xây dựng có thể học đặc trưng hình ảnh, nhưng để cải thiện hiệu năng, cần áp dụng Transfer Learning từ các mô hình khác, giúp tăng độ chính xác và khả năng khái quát hóa đáng kể.

Chương 5 : So sánh giữa các mô hình

5.1 Tổng hợp kết quả huấn luyện

Dưới đây tổng hợp kết quả của ba mô hình sau khi huấn luyện và đánh giá trên cùng bộ dữ liệu kiểm thử (Oxford 102 Flower Dataset).

Mô hình	Accuracy	Precision	Recall	F1-Score	Loss
CNN cải tiến	0.5543	0.5742	0.5133	0.5104	1.7662
MobileNetV2	0.9194	0.9100	0.9100	0.9100	0.2970
EfficientNetB0	0.9328	0.9298	0.9151	0.9152	0.1862

Nhận xét sơ bộ:

- Mô hình EfficientNetB0 cho hiệu năng cao nhất trên tất cả các chỉ số.
- MobileNetV2 cũng đạt kết quả rất tốt, chỉ thấp hơn EfficientNetB0 khoảng 1–1.5%.
- CNN cải tiến có độ chính xác thấp hơn đáng kể (55%), đóng vai trò baseline để đối chiếu.

5.2 So sánh chi tiết quá trình huấn luyện

Tiêu chí	CNN cải tiến	MobileNetV2	EfficientNetB0	Nhận xét
Độ chính xác (Accuracy)	Thấp (55%)	Cao (92%)	Rất cao (93%)	EfficientNetB0 vượt trội
Khả năng khái quát hóa (Generalization)	Kém	Tốt	Rất tốt	Transfer learning giúp học đặc trưng sâu

Mức độ hội tụ (Convergence)	Chậm	Nhanh, ổn định	Nhanh nhất, ổn định nhất	EfficientNet tối ưu hơn
Khả năng chống overfitting	Trung bình	Tốt	Rất tốt	EfficientNet kiểm soát tốt hơn nhờ batchnorm và scaling
Chi phí tính toán	Thấp	Trung bình	Cao hơn một chút	CNN nhẹ nhưng yếu về hiệu năng

CNN cải tiến:

Độ chính xác tăng chậm và dừng lại ở mức 55%, loss giảm nhưng dao động nhiều mô hình hội tụ chậm, dễ bị ảnh hưởng bởi nhiễu dữ liệu.

=>Biểu hiện của kiến trúc đơn giản, chưa đủ độ sâu và khả năng trích xuất đặc trưng mạnh.

MobileNetV2:

Accuracy tăng nhanh, hội tụ ổn định sau 10 epoch đầu. Validation Accuracy 92% cho thấy transfer learning từ ImageNet giúp mô hình học nhanh và tổng quát tốt.

=>Biểu đồ loss giảm nhanh và duy trì ổn định, không xuất hiện overfitting.

EfficientNetB0:

Đường học (learning curve) của EfficientNetB0 rất ổn định: Train và Val Accuracy gần như trùng nhau (93%), loss giảm nhanh và duy trì thấp (0.18).

=>Thể hiện khả năng cân bằng giữa độ chính xác và khái quát hóa của kiến trúc EfficientNet.

5.3. So sánh ma trận nhầm lẫn giữa các mô hình

Ma trận nhầm lẫn (Confusion Matrix) phản ánh trực quan khả năng phân loại đúng – sai giữa các lớp của mô hình, giúp nhận diện các nhóm dữ liệu mà mô hình còn nhầm lẫn. Ba hình dưới đây lần lượt thể hiện kết quả của các mô hình **CNN cải tiến**, **MobileNetV2**, và **EfficientNetB0** trên cùng tập kiểm thử gồm 102 lớp hoa.

Mô hình	Mức độ nhầm lẫn	Độ tập trung trên đường chéo	Nhận xét tổng quát
CNN cải tiến	Cao – nhiều điểm sai rải rác	Thấp – đường chéo mờ, loang lổ	Mô hình học được đặc trưng cơ bản nhưng chưa tách lớp tốt.
MobileNetV2	Trung bình – nhầm ở vài nhóm tương đồng	Cao – đường chéo rõ nét	Học tốt đặc trưng tổng quát, giảm sai sót đáng kể.
EfficientNetB0	Rất thấp – gần như không nhầm	Rất cao – đường chéo sáng rõ, gọn	Phân loại chính xác, mô hình hội tụ mạnh và ổn định.

Kết luận:

- Sự khác biệt giữa ba ma trận nhầm lẫn phản ánh rõ ràng hiệu quả của Transfer Learning.
- CNN cải tiến còn nhiều sai sót → chưa đủ khả năng học đặc trưng phức tạp.

- MobileNetV2 và EfficientNetB0 đã tận dụng tốt tri thức học sẵn, giúp mô hình phân biệt chính xác gần như toàn bộ các lớp hoa, trong đó EfficientNetB0 vượt trội nhất cả về độ chính xác lẫn tính ổn định.

5.4. Kết luận

Qua kết quả thực nghiệm, có thể rút ra các nhận định sau:

- Transfer Learning (**MobileNetV2**, **EfficientNetB0**) mang lại hiệu quả vượt trội so với CNN truyền thống.
- **EfficientNetB0** là mô hình tối ưu nhất trong thí nghiệm này với Accuracy = 93.28%, F1 = 0.9152, Loss = 0.1862, thể hiện khả năng học và khái quát hóa tốt nhất.
- **MobileNetV2** vẫn là lựa chọn phù hợp khi muốn cân bằng giữa hiệu năng và tốc độ.
- CNN cải tiến chỉ đạt 55%, thể hiện vai trò baseline nhưng vẫn hữu ích để chứng minh lợi ích rõ ràng của phương pháp Transfer Learning.

Trong bài toán phân loại 102 loài hoa, **EfficientNetB0** cho thấy ưu thế vượt trội về độ chính xác, khả năng tổng quát và tính ổn định.

Điều này chứng minh hiệu quả của việc kết hợp tri thức tiền huấn luyện cùng fine-tuning có kiểm soát, giúp mô hình đạt hiệu suất cao ngay cả với tập dữ liệu có quy mô vừa phải.

Chương 6: Hướng phát triển

Sau khi hoàn thành quá trình nghiên cứu và xây dựng hệ thống nhận dạng hình ảnh các loài hoa dựa trên các mô hình học sâu, nhóm đã đạt được những kết quả khả quan, đặc biệt là với các mô hình **Transfer Learning** như **MobileNetV2** và **EfficientNetB0**. Tuy nhiên, hệ thống vẫn còn nhiều tiềm năng để cải thiện và mở rộng trong tương lai.

Dưới đây là một số hướng phát triển chính mà nhóm đề xuất:

6.1. Nâng cao hiệu năng mô hình

Tăng cường dữ liệu (Data Augmentation nâng cao):

Áp dụng các kỹ thuật như CutMix, MixUp, Random Erasing, Color Jitter, Gaussian Noise để tạo ra tập dữ liệu huấn luyện đa dạng hơn, giúp mô hình khái quát tốt hơn với ảnh thực tế.

Fine-tuning sâu hơn trên mô hình pretrained:

Thay vì chỉ huấn luyện phần đầu ra, có thể mở khóa nhiều lớp convolution sâu hơn của **EfficientNet** hoặc MobileNet để tinh chỉnh trọng số phù hợp hơn với đặc trưng của bộ dữ liệu hoa.

Sử dụng kiến trúc tiên tiến hơn:

Thử nghiệm các mô hình mới như **EfficientNetV2**, **ConvNeXt**, **Vision**

Transformer (ViT) hoặc **Swin Transformer**, vốn cho hiệu năng vượt trội trong các bài toán thị giác máy tính gần đây.

Tối ưu tham số huấn luyện (Hyperparameter Tuning):

Áp dụng các kỹ thuật như Grid Search, Bayesian Optimization hoặc Optuna để tìm ra cấu hình tối ưu cho Learning Rate, Batch Size, Dropout, Optimizer,...

6.2 Mở rộng phạm vi ứng dụng

Triển khai thành ứng dụng thực tế:

Xây dựng giao diện Web hoặc Mobile App cho phép người dùng tải lên ảnh hoa và nhận dạng trực tiếp. Có thể dùng **TensorFlow Lite**, **ONNX Runtime** hoặc **FastAPI** để triển khai.

Phát triển hệ thống tra cứu thông tin loài hoa:

Sau khi nhận dạng, hệ thống có thể hiển thị tên hoa, nguồn gốc, khu vực sinh trưởng, ý nghĩa biểu tượng, hình ảnh tương tự,... tạo thành ứng dụng học tập và nghiên cứu sinh học.

Tích hợp với công nghệ IoT / camera thông minh:

Áp dụng mô hình vào các thiết bị camera trong nhà kính hoặc vườn hoa, giúp tự động nhận dạng, giám sát và thống kê số lượng, trạng thái sinh trưởng của từng loài.

Phát triển API mở (Flower Recognition API):

Cho phép các nhà phát triển khác tích hợp mô hình nhận dạng hoa vào ứng dụng riêng (ví dụ: giáo dục, du lịch, nghệ thuật số).

6.3 Cải thiện chất lượng dữ liệu

Tăng quy mô và đa dạng bộ dữ liệu:

Kết hợp Oxford 102 Flower Dataset với các bộ dữ liệu khác (như Flowers Recognition Dataset hoặc Flower Species Kaggle Dataset) để nâng số lượng mẫu và

độ phong phú của đặc trưng hình ảnh.

Tiền xử lý nâng cao:

Áp dụng các kỹ thuật như cân bằng histogram (Histogram Equalization), khử nhiễu (Denoising), và phát hiện biên (Edge Detection) nhằm cải thiện chất lượng đầu vào trước khi huấn luyện.

Tạo tập dữ liệu phân tầng (Stratified Sampling):

Đảm bảo mỗi lớp hoa có số lượng mẫu cân bằng để giảm sai lệch (bias) trong huấn luyện.

Chương 7: Tổng kết

Trong đề tài này, nhóm đã tiến hành nghiên cứu, xây dựng và đánh giá các mô hình học sâu phục vụ cho bài toán phân loại hình ảnh 102 loài hoa. Bắt đầu từ mô hình CNN cải tiến tự xây dựng, nhóm từng bước áp dụng các mô hình Transfer Learning như **MobileNetV2** và **EfficientNetB0**, nhằm so sánh hiệu năng, khả năng học đặc trưng và mức độ khái quát hóa của từng mô hình trên cùng tập dữ liệu.

Kết quả thực nghiệm cho thấy, **EfficientNetB0** đạt hiệu suất cao nhất với độ chính xác 93.28%, Precision 92.98%, Recall 91.51% và F1-Score 91.52%, khẳng định ưu thế vượt trội của phương pháp Transfer Learning kết hợp Fine-Tuning so với việc huấn luyện mô hình từ đầu. **MobileNetV2** cũng đạt hiệu quả rất tốt (Accuracy 91.94%), đồng thời có ưu điểm về tốc độ và độ gọn nhẹ, phù hợp cho triển khai trên thiết bị di động hoặc hệ thống giới hạn tài nguyên. Trong khi đó, mô hình CNN cải tiến chỉ đạt khoảng 55.43%, song vẫn đóng vai trò quan trọng như một mô hình nền tảng (baseline) giúp minh chứng rõ ràng cho lợi ích của việc sử dụng các mô hình học sâu đã được huấn luyện trước.

Thông qua quá trình thực hiện, nhóm đã hiểu rõ quy trình CRISP-DM, bao gồm các giai đoạn: thu thập, tiền xử lý, huấn luyện, đánh giá và triển khai mô hình. Các kỹ thuật như tăng cường dữ liệu (**Data Augmentation**), chuẩn hóa đặc trưng (Normalization) và sử dụng Callback (**EarlyStopping**, **ModelCheckpoint**) đã góp phần quan trọng vào việc nâng cao độ ổn định và chất lượng của mô hình.

Đề tài không chỉ đạt được mục tiêu về mặt kỹ thuật mà còn mang ý nghĩa thực tiễn cao, khi hoàn toàn có thể mở rộng để xây dựng ứng dụng nhận dạng hoa tự động phục vụ học tập, nghiên cứu hoặc trong lĩnh vực nông nghiệp thông minh, giáo dục sinh học và du lịch sinh thái.

Tuy vậy, hệ thống vẫn còn những giới hạn nhất định, đặc biệt là về độ đa dạng của tập dữ liệu và khả năng nhận dạng trong điều kiện thực tế (ánh sáng, góc chụp, nhiễu).

ảnh). Trong tương lai, việc mở rộng quy mô dữ liệu, áp dụng các mô hình tiên tiến hơn (như **EfficientNetV2**, **ViT**, **Swin Transformer**) và triển khai thực tế qua Web/App sẽ là hướng đi đầy tiềm năng.

Tóm lại, đề tài đã hoàn thành trọn vẹn các mục tiêu nghiên cứu ban đầu, xây dựng thành công hệ thống phân loại ảnh hoa với độ chính xác cao, đồng thời chứng minh hiệu quả rõ rệt của Transfer Learning trong lĩnh vực Khai phá dữ liệu hình ảnh (Image Data Mining). Kết quả đạt được là nền tảng quan trọng cho các nghiên cứu tiếp theo và mở ra nhiều hướng ứng dụng rộng rãi trong tương lai.

Tham khảo:

[Oxford 102 Flowers Classification using a Neural Network Classifier](#)

[oxford 102 flowers classification, deep learning image classification tutorial](#)